

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE



“SISTEMA RECOMENDADOR PARA UNA APLICACIÓN TRANSACCIONAL EN LA INDUSTRIA DE DISTRIBUCIÓN DE COMBUSTIBLES”

MACARENA ALEXANDRA ANDRADE OYARZÚN

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: José Luis Martí Lara
Profesor Correferente: Pedro Francisco Godoy Barrera

Marzo - 2023

DEDICATORIA

Dedicado a mi abuelito, quien mientras estuvo en vida, creyó en mis capacidades cuando ni yo misma era capaz de hacerlo, espero estés viéndome orgulloso desde el cielo.

AGRADECIMIENTOS

Primero que todo quiero agradecer a mi familia; a mis padres, mi hermana y mi abuela, por haberme proveído los recursos necesarios para poder estudiar y apoyarme a lo largo de toda la carrera.

En el transcurso de estos años en la universidad conocí muchas personas que fueron parte importante de mi desarrollo. Quiero agradecer a Bastián, quien fue la persona que me acompañó la mayor parte este proceso, gracias por todo tu apoyo y contención. Gracias a Gaspar y Sebastián, quienes fueron mis compañeros de estudio y tareas en mis primeros años en la carrera. Gracias a mis amigos; Pedro, Matías, Fabián, John, Daniel, Javiera, Braulio, Farid y Cristóbal, por hacer mi estadía en la universidad un poco más amena. Gracias a Sebastián, Ignacio y Gerardo, por hacer de la Feria de Software una lindo recuerdo y por el compañerismo entregado hasta el día de hoy. Gracias a la profesora Cecilia Reyes y al profesor Luis Hevia, que en múltiples ocasiones me dieron oportunidades para poder desempeñarme como ayudante y coordinadora, lo que ayudó a potenciar mi crecimiento personal.

El año pasado se me dio la oportunidad de trabajar en este proyecto, el cual me abrió las puertas al mundo laboral, muchas gracias Lukas por confiar ciegamente en mis capacidades y guiarme a lo largo de este proceso. Gracias a Felipe, José y Camila, quienes desde su experiencia me ayudaron con sus conocimientos técnicos. Finalmente, gracias al profesor José Luis Martí, quien amablemente acogió este proyecto y siempre tuvo una buena disposición para apoyar en el desarrollo de este.

RESUMEN

Resumen— El presente documento tiene como objetivo explicar el trabajo realizado para el desarrollo de un sistema recomendador de productos personalizados para una aplicación móvil transaccional. Para lograrlo se hizo un estudio con el objetivo de entender las necesidades del negocio y conocer la factibilidad del proyecto, se analizaron distintas técnicas para generar recomendaciones y se probaron distintas formas de modelar las transacciones de los usuarios. Se realizó un workshop para saber si las recomendaciones personalizadas son del gusto de los usuarios de la aplicación, donde la gran mayoría de las recomendaciones lograron satisfacer las necesidades de los usuarios. Como el sistema recomendador fue capaz de brindar satisfacción a los usuarios de prueba, se espera que la implementación del sistema en la aplicación pueda llegar a generar un impacto positivo en la experiencia de los usuarios activos y llegar a potenciar el uso de la aplicación.

Palabras Clave— Sistema Recomendador; Personalización; E-Commerce.

ABSTRACT

Abstract— Corresponde a la traducción al idioma inglés del Resumen anterior. Sujeto a la misma regla de extensión del Resumen.

Keywords— Corresponde a la traducción al idioma inglés de Palabras Clave anteriores.

GLOSARIO

Aquí se deben colocar las siglas mencionadas en el trabajo y su explicación, por orden alfabético. Por ejemplo:

DI: Departamento de Informática.

UTFSM: Universidad Técnica Federico Santa María.

ÍNDICE DE CONTENIDOS

RESUMEN	IV
ABSTRACT	IV
GLOSARIO	V
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS	VIII
INTRODUCCIÓN	1
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	2
1.1 CONTEXTO	2
1.2 OBJETIVOS	3
OBJETIVO GENERAL	3
OBJETIVOS ESPECÍFICOS	3
CAPÍTULO 2: MARCO CONCEPTUAL	5
2.1 SISTEMAS RECOMENDADORES	5
2.1.1 SISTEMAS RECOMENDADORES BASADOS EN CONTENIDO	6
2.1.2 SISTEMAS RECOMENDADORES DE FILTRADO COLABORATIVO	6
2.1.2.1 MÉTODOS BASADOS EN MEMORIA	6
2.1.2.2 MÉTODOS BASADOS EN MODELOS	7
FACTORIZACIÓN MATRICIAL	8
SVD	8
ALS	10
2.2 ANÁLISIS DE CANASTA	11
2.3 METODOLOGÍAS PARA PROYECTOS DE ANÁLISIS DE DATOS	12
2.3.1 CRISP-DM	12
2.3.2 KDD	14
2.3.3 SEMMA	15
2.3.4 DST	16
CAPÍTULO 3: PROPUESTA DE SOLUCIÓN	19
3.1 Metodología de trabajo	19
3.2 Herramientas utilizadas	20
3.3 Desarrollo del Proyecto	20
3.3.1 Entendimiento del negocio	21
3.3.2 Entendimiento de los datos	23
Extracción de datos	23
Análisis Exploratorio de Datos	25
3.3.3 Preparación de los datos	38

	Extracción de datos	38
	Filtrado de datos	39
	Transformación de datos	39
3.3.4	Modelamiento	40
CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN		41
4.1	Desarrollo del Proyecto	41
4.1.1	Modelamiento	41
	Ingeniería de características	41
	Implementación de Alternating Least Square (ALS)	42
4.1.2	Evaluación	43
	Evaluación Cuantitativa: Métrica de desempeño	43
	Evaluación Cualitativa: Pruebas en usuarios reales	44
4.1.3	Despliegue	50
	Casos de uso	50
CONCLUSIONES		55
ANEXOS		58
REFERENCIAS BIBLIOGRÁFICAS		64

ÍNDICE DE FIGURAS

1	Ilustración simplificada del enfoque de factores latentes.	8
2	Fases de la metodología CRISP-DM.	13
3	Fases de la metodología KDD.	14
4	Fases de la metodología SEMMA.	15
5	Mapa con las actividades de DST.	17
6	Página de inicio de Muevo.	22
7	Página de inicio de Escanea y Paga.	22
8	Productos con mayor aparición en boletas.	25
9	Grupos con mayor aparición en boletas.	26
10	Distribución de transacciones por momento del día.	27
11	Distribución de transacciones por tiempo.	28
12	Distribución de transacciones por formato de tienda.	28
13	Productos con mayor aparición en boletas en horario de mañana.	29
14	Productos con mayor aparición en boletas en horario tarde.	30
15	Productos con mayor aparición en boletas en horario de noche.	30
16	Productos con mayor aparición en boletas en horario de madrugada.	31
17	Productos con mayor aparición en boletas en lluvias.	32
18	Productos con mayor aparición en boletas en cielos cubiertos.	32
19	Productos con mayor aparición en boletas en cielos nubosos.	33
20	Productos con mayor aparición en boletas en intervalos nubosos.	33
21	Productos con mayor aparición en boletas en cielos despejados.	34
22	Productos con mayor aparición en boletas en carretera.	34
23	Productos con mayor aparición en boletas en ciudad.	35
24	Set final de entrenamiento.	42
25	Encuesta sobre el sentido del orden de categorías personalizado.	46
26	Encuesta sobre preferencia sobre el orden de categorías personalizado.	46
27	Encuesta sobre el sentido del orden de productos personalizado.	47
28	Encuesta sobre la preferencia sobre el orden de productos personalizado.	47
29	Encuesta sobre Recomendaciones de Productos Sustitutos.	48
30	Encuesta sobre Recomendaciones de Productos Complementarios.	48
31	Encuesta sobre Recomendaciones Personalizadas.	49
32	Encuesta sobre Recomendaciones de Usuarios Similares.	50
33	Cambios en la interfaz dado el orden personalizado de categorías.	51
34	Cambios en la interfaz dado el orden personalizado de productos.	52
35	Cambios en la interfaz para recomendaciones de productos sustitutos y complementarios.	53
36	Cambios en la interfaz para recomendaciones personalizadas y de usuarios similares.	54

ÍNDICE DE TABLAS

1	Fases de CRISP-DM.	19
2	Bases de datos de Copec.	23
3	Tablas utilizadas para el EDA.	24
4	Análisis de Canasta de Carretera-Mañana.	36
5	Análisis de Canasta de Carretera-Tarde.	36
6	Análisis de Canasta de Carretera-Noche.	36
7	Análisis de Canasta de Carretera-Madrugada.	36
8	Análisis de Canasta de Ciudad-Mañana.	36
9	Análisis de Canasta de Ciudad-Tarde.	36
10	Análisis de Canasta de Ciudad-Noche.	37
11	Análisis de Canasta de Ciudad-Madrugada.	37
12	Datos preprocesados.	38
13	Datos procesados.	38

INTRODUCCIÓN

Con el surgimiento del comercio electrónico o *E-Commerce*, muchas empresas encontraron una oportunidad más allá de simplemente vender productos por medios digitales, también descubrieron un nuevo canal por el cual conocer mejor a sus clientes. Los datos extraídos por estos canales ayudan a conocer qué tipos de productos le gustan a las personas, cuál es el recorrido que hacen para comprar un producto y qué los detiene a completar una compra. Por otra parte, Copec, empresa chilena que surgió con el objetivo de distribuir combustible para fomentar el desarrollo del país, actualmente busca expandir sus servicios para poder brindar experiencias que entreguen comodidad a sus consumidores. Bajo esta premisa es que los equipos de Marketing y de Productos Digitales buscan desarrollar campañas y productos que entreguen valor a sus clientes, pero que de igual forma puedan recibir valor de vuelta al recopilar información a partir de interacción de los clientes con los servicios utilizados. De esta forma, es que el equipo de Productos Digitales busca implementar un sistema recomendador de productos para personalizar el E-Commerce de la aplicación Muevo de Copec donde, por medio de las transacciones hechas por los usuarios, el sistema recomendador aprenderá a identificar los intereses de los clientes, y así poder recomendarles productos que lleguen a ser de su agrado y mejorar su experiencia dentro de la aplicación. Este proyecto será una prueba de concepto, por lo cual el levantamiento del sistema recomendador en la aplicación no será parte de este trabajo.

En el primer capítulo de este informe se contextualiza acerca de Copec, el equipo de Productos Digitales y sus necesidades actuales; además, se definen los objetivos generales y específicos del proyecto. En el segundo capítulo se explica el marco conceptual del proyecto, abarcando los tipos de sistemas recomendadores y las metodologías de proyectos de ciencia de datos. En el tercer capítulo se presenta la propuesta de solución, donde se selecciona una metodología de trabajo y se explica la planificación del proyecto. En el cuarto capítulo se explica el trabajo realizado a partir de la planificación previamente planteada. Por último, se presentan las conclusiones con los resultados obtenidos a partir del desarrollo del proyecto, señalando los aprendizajes obtenidos y posibles trabajos a futuro.

CAPÍTULO 1

DEFINICIÓN DEL PROBLEMA

En la primera sección de este capítulo se contextualiza acerca del origen de la empresa Copec y sus subsidiarias, se explica el funcionamiento de la aplicación Muevo y se presenta a los equipos que trabajan en ella, para finalmente identificar las necesidades que tienen actualmente. En la segunda sección se presentan los objetivos a resolver para solucionar el problema de la empresa.

1.1. CONTEXTO

Copec es una empresa fundada el año 1934 que nació con el objetivo de asegurar el suministro de combustibles en el país [Copec, 1970]. A finales de 1936, contaba con estaciones de servicio para el suministro de combustible desde la región de Coquimbo hasta Magallanes. Posteriormente Copec decidió expandir su negocio y orientar su servicio a sus clientes; es así como en el año 1990, se inaugura su red de tiendas de conveniencia, "Pronto". En 1997, la red de locales "Pronto" sería reforzada con tecnología y nuevos servicios. En el año 2001, nace la red de tiendas "Punto", cuyo objetivo es cubrir las necesidades del cliente que se encuentra en movimiento durante el día [Punto, 2001]. En 2018, Copec lanza la red "Voltex", destinada a la electromovilidad, la cual cuenta con cargadores rápidos en 23 estaciones de servicio que conectan 700 km lineales, convirtiéndola en la red más larga de Latinoamérica. En el año 2020, Copec lanza la aplicación Muevo, con el objetivo de brindar una nueva experiencia a sus clientes.

La aplicación Muevo permite realizar pagos sin contacto, evitando el uso de tarjetas o dinero en efectivo. Para poder hacer uso de esta funcionalidad es necesario tener vinculada una tarjeta de crédito o débito en la aplicación; para realizar el pago se escanea el código de barras en un tótem de servicio y el cargo se realiza automáticamente [Muevo, 2020]. Esta funcionalidad está actualmente habilitada para pagos de combustible en Estaciones de Servicio, cargas en la red "Voltex", lubricantes en "Lub" y productos en tiendas "Punto" y "Pronto"; a estos se les referirá de aquí en adelante como servicios de Copec. Dentro de la aplicación se puede encontrar el programa de beneficios "Full Copec", en donde se acumulan puntos al hacer compras en los servicios de Copec, los cuales pueden ser canjeados por distintos productos dentro del catálogo que "Full Copec" ofrece en la aplicación. Próximamente está por implementarse la funcionalidad "Escanea y Paga", la cual permite pagar productos desde la aplicación sin la necesidad de pasarlos por la caja registradora.

Dentro de Copec existen equipos de trabajos destinados a detectar requerimientos en Muevo para llevarlos a los equipos de desarrollo de la aplicación; a estos equipos de trabajo se les conoce como Pilares. Los Pilares se dividen en 4 áreas: Plataforma, E-Commerce, Fidelidad y Estaciones de Servicio. Actualmente, los Pilares buscan cumplir con ciertos objetivos,

entre ellos se encuentra aumentar el número de clientes activos en Muevo, potenciar el uso de la aplicación para realizar compras y mejorar la experiencia en tiendas. El cumplimiento de estos objetivos se puede evaluar con el incremento de distintos indicadores. Uno de los indicadores es el *Monthly Active Users* o MAU, el cual mide la cantidad de usuarios activos mensuales en la aplicación, donde se considera usuario activo a aquel que haya hecho al menos una transacción al mes. Se busca incrementar el MAU de 670.000 a 1.200.000 a diciembre de 2022. Otro indicador que se busca aumentar es el ticket promedio, el cual mide el monto promedio por boleta en tiendas Punto y Pronto.

El Pilar de *E-Commerce* espera que, al mejorar la experiencia de usuario en la aplicación, sea posible aumentar el ticket promedio e, indirectamente, el MAU, por lo que propone implementar un sistema recomendador de productos en el carro de compras de la funcionalidad “Escanea y Paga”. Se espera que, al recomendar productos personalizados, en caso de que estos lleguen a ser del gusto del cliente, este podría llegar a comprar más productos de los que hubiese adquirido si no hubiese recibido la recomendación, lo que provocaría un aumento en el ticket promedio. Se tiene la hipótesis de que, si el cliente se siente a gusto con la recomendación, mejoraría su experiencia en la aplicación y podría llegar a usarla con mayor frecuencia, y de ocurrir esto, se podría llegar a impactar al indicador MAU.

1.2. OBJETIVOS

OBJETIVO GENERAL

Desarrollar un sistema recomendador de productos para Muevo utilizando los datos transaccionales de los clientes en tiendas Punto y Pronto de Copec, para aumentar el ticket promedio de estas tiendas.

OBJETIVOS ESPECÍFICOS

- Analizar metodologías que permitan desarrollar un sistema recomendador de buena calidad mediante el seguimiento de buenas prácticas, para seleccionar la más adecuada.
- Analizar los algoritmos más utilizados al desarrollar sistemas recomendadores, para escoger los más convenientes para resolver problemas similares.
- Desarrollar un sistema recomendador de calidad y evaluar su desempeño en base a métricas como RMSE, F1 Score y/o MAP, para asegurar que las recomendaciones sean adecuadas para los usuarios.

- Validar que el sistema recomendador desarrollado cumpla con la aprobación de áreas comerciales.

CAPÍTULO 2

MARCO CONCEPTUAL

En la primera sección de este capítulo se define lo que son los sistemas recomendadores y se explican los distintos tipos y métodos que existen, haciendo mayor énfasis en los sistemas recomendadores de filtrado colaborativo. En la segunda sección se abordan distintas metodologías de trabajo para proyectos de minería de datos y ciencia de datos, explicando sus fases y usos.

2.1. SISTEMAS RECOMENDADORES

Los sistemas recomendadores son softwares o técnicas que recomiendan objetos al usuario [Ricci *et al.*, 2022]. Estas recomendaciones están destinadas a ayudar al usuario a tomar decisiones. Las recomendaciones suelen ser útiles para usuarios que no cuentan con experiencia en el sistema, ya que los orienta a elegir entre las múltiples opciones que este ofrece. Existen tanto recomendaciones personalizadas como no personalizadas. Las recomendaciones personalizadas son útiles para usuarios específicos o segmentos de usuarios, ya que reciben recomendaciones específicas para satisfacer sus intereses. Por otra parte, las recomendaciones no personalizadas sirven para usuarios nuevos, ya que no se cuenta con información acerca de ellos, y en su forma más sencilla recomiendan los objetos más populares del sistema.

Los sistemas recomendadores más sencillos muestran una lista de objetos, los que se encuentran ordenados de acuerdo con qué tanto se adecúan al usuario, considerando sus preferencias y limitaciones. Para poder determinar cuales son los objetos más adecuados, el sistema recolecta las preferencias del usuario, donde puede recopilar información de manera directa o indirecta. El sistema puede recolectar información de manera directa mediante la valoración del objeto por parte del usuario, y de manera indirecta interpretando las acciones del usuario en el sistema, donde la navegación en la página de un objeto en particular puede ser un indicio de preferencia por parte del usuario.

Con el surgimiento de sitios web dedicados al comercio digital, más conocidos como *E-commerce*, surgió la necesidad de brindar recomendaciones a los usuarios, ya que estos pueden sentirse abrumados al encontrarse con un amplio rango de objetos, y al no contar con conocimiento previo, puede provocar que el usuario tome una decisión que no resulte ser de su agrado. El exceso de opciones puede llegar a brindar una mala experiencia en lugar de volverse algo atractivo para los usuarios.

Existen distintos tipos de sistemas recomendadores, los cuales difieren en su conocimiento del dominio, los datos utilizados y en sus algoritmos. A continuación, se explican los distintos tipos de sistemas recomendaciones y los métodos que estos utilizan.

2.1.1. SISTEMAS RECOMENDADORES BASADOS EN CONTENIDO

El sistema recomienda objetos basados en otros similares que al usuario le hayan gustado en el pasado [Ricci *et al.*, 2022]. Los objetos se caracterizan a sí mismos por su contenido, el cual se transforma a un formato de vector, donde cada valor representa una característica del objeto. Dos objetos son similares si cuentan con múltiples características similares, por ejemplo, si a un usuario le gusta una canción del género pop, el sistema le recomendará canciones del mismo género musical.

Este tipo de sistemas cuenta con ciertas desventajas. Existen casos en que los objetos no cuentan con suficiente contenido que los caracterice, lo que provoca que el sistema no recomiende estos objetos. Además, como se busca recomendar objetos similares entre sí, el sistema falla en identificar objetos diferentes de los que fueron previamente consumidos, evitando que se recomiende un objeto que potencialmente podría llegar a ser de interés para el usuario.

2.1.2. SISTEMAS RECOMENDADORES DE FILTRADO COLABORATIVO

El sistema, para recomendar objetos nuevos al usuario, busca otros usuarios con intereses similares al suyo, donde la similitud entre estos se puede ver reflejada en la forma en que estos valoran los objetos [Ricci *et al.*, 2022]. Por ejemplo, si el usuario valoró positivamente un libro de la saga de “Harry Potter”, el sistema buscará a otros usuarios que también hayan disfrutado de aquel libro, para así recomendar algún libro que haya sido valorado positivamente por estos usuarios.

Los sistemas recomendadores de filtrado colaborativo cubren las falencias de los sistemas basados en contenido. Los objetos que no cuentan con suficiente contenido para caracterizarse a sí mismos pueden llegar a ser recomendados si es que cuentan con valoraciones previas por parte de otros usuarios. Además, estos sistemas pueden recomendar objetos que cuentan con características distintas que los objetos consumidos previamente por el usuario, si es que estos cuentan con valoraciones de otros usuarios que cuenten con un historial de valoraciones similares. El filtrado colaborativo no busca asegurar la calidad de un objeto mediante su contenido, sino en las valoraciones de los usuarios.

2.1.2.1. MÉTODOS BASADOS EN MEMORIA

Los sistemas recomendadores basados en memoria almacenan las valoraciones hechas por los usuarios hacia los objetos, de forma que en base a estas puntuaciones, se pueda predecir la valoración para nuevos objetos [Ricci *et al.*, 2022]. Este método se puede dividir en los basados por usuarios y los basados en objetos. Los basados en usuarios evalúan el interés de un usuario por un objeto, en base a las valoraciones de otros usuarios similares hacia el

mismo objeto, donde un usuario se considera similar a otro si tienen patrones de valoración similares. Por otra parte, los métodos basados en objetos buscan evaluar el interés de un usuario por un objeto basándose en las valoraciones de otros usuarios a otros objetos similares, donde se consideran objetos similares a aquellos que reciben valoraciones similares por parte de un mismo usuario.

Algunas de las ventajas de los métodos basados en vecindarios son las siguientes:

- **Simplicidad:** Estos métodos son intuitivos y fáciles de implementar. En su forma más sencilla, el único parámetro que se debe ajustar es la cantidad de vecinos usados en la predicción.
- **Justificabilidad:** Las recomendaciones tienen una justificación concisa e intuitiva. Por ejemplo, en los vecindarios basados en objetos, la recomendación es justificada por la lista de objetos del vecindario y por las valoraciones de los usuarios sobre los objetos.
- **Eficiencia:** Estos métodos no requieren costosas fases de entrenamiento, a diferencia de los métodos basados en modelos, los cuales deben realizar esta fase de manera frecuente en aplicaciones comerciales. Si bien generar la recomendación es un proceso más costoso que para un método basado en modelos, se puede pre-calcular el vecino más cercano de manera offline, de forma que, cuando el sistema vuelve a estar online, la recomendación se obtiene automáticamente. Almacenar los vecinos no requiere de tanta memoria, por lo que permite que este proceso sea escalable para aplicaciones con millones de usuarios y objetos.
- **Estabilidad:** Estos métodos no suelen verse altamente afectados al agregar nuevos usuarios, objetos o valoraciones al sistema. Cuando la similitud entre objetos es calculada, el sistema estará listo para hacer la recomendación.

2.1.2.2. MÉTODOS BASADOS EN MODELOS

Los sistemas recomendadores basados en modelos, a diferencia de los basados en memoria que generan predicciones en base a las valoraciones almacenadas, buscan descubrir factores latentes que representen las características que describen a los usuarios y a los objetos, de forma que el modelo pueda entender el razonamiento detrás de las valoraciones observadas [Ricci *et al.*, 2022]. Estos modelos se entrenan con los datos existentes de valoraciones hechas por los usuarios, de forma que se pueda predecir cómo un usuario valorará un nuevo objeto. En la figura 1 se puede observar un diagrama simplificado sobre cómo se representan los usuarios y películas en un plano de factores latentes.

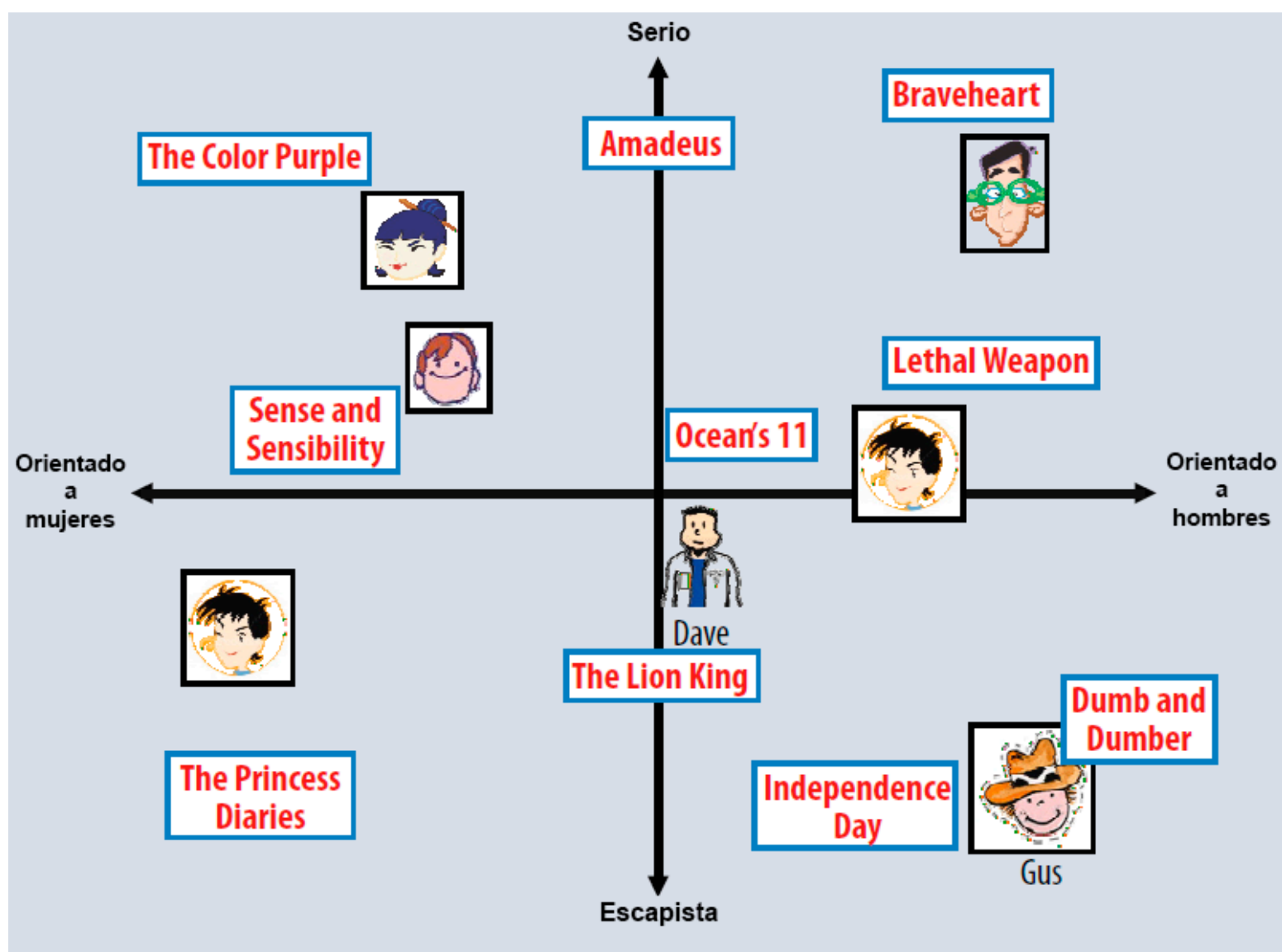


Figura 1: Ilustración simplificada del enfoque de factores latentes.
Fuente: [Koren et al., 2009]

FACTORIZACIÓN MATRICIAL

La factorización matricial es uno de los métodos más exitosos para sistemas recomendadores basados en métodos de factores latentes. La factorización matricial busca caracterizar a los usuarios y objetos mediante vectores, donde los valores corresponden a factores que influyen en la valoración de un objeto [Koren et al., 2009]. Así, si un usuario y un objeto comparten una alta correspondencia entre sus factores, entonces el sistema recomendará aquel objeto al usuario. Este método se caracteriza por su buena escalabilidad y exactitud.

SVD

Singular Value Decomposition es un algoritmo de reducción de dimensionalidad, el cual propone que una matriz A se puede descomponer de la forma $A = U\lambda V^T$, donde A es una matriz de $m \times n$, U es una matriz de $n \times r$, λ es una matriz diagonal de $r \times r$ y V es una matriz de $m \times r$, donde r corresponde al número de características. Bajo esta premisa, surgen los modelos basados en SVD, los cuales consisten en factorizar la matriz que almacena las valo-

raciones hechas por los usuarios hacia los productos, de forma que al separar las matrices, se obtengan dos matrices, una de usuarios y otra de productos, donde los valores representarán los factores latentes que caracterizan a cada usuario y a cada producto respectivamente [Koren *et al.*, 2009] Cada objeto i está asociado a un vector q_i y cada usuario u está asociado a un vector p_u . Para el objeto i , los valores del vector q_i miden qué tanto se asocia el objeto a cierto factor, ya sea de manera positiva o negativa. En el caso del usuario u , el vector p_u mide qué tanto interés tiene el usuario con respecto a cada factor, ya sea positiva o negativamente. La relación entre el usuario y el objeto se puede ver representada mediante la operación $q_i^T p_u$, cuyo resultado representa qué tanto le interesa al usuario u las características del objeto i . Así, la expresión que define la predicción de la valoración del usuario u hacia el objeto i es la siguiente:

$$\hat{r}_{ui} = q_i^T p_u \quad (1)$$

Los valores de las matrices se calculan por medio de gradiente descendiente estocástico, el cual es un algoritmo de aprendizaje, donde por cada iteración se recalculan los vectores de las matrices para lograr llegar al mínimo de una función. De esta forma, se calcula el error:

$$e_{ui} = r_{ui} - q_i^T p_u \quad (2)$$

para posteriormente recalcular los vectores q_i y p_u :

$$q_i \leftarrow q_i + \gamma(e_{ui}p_u - \lambda q_i) \quad (3)$$

$$p_u \leftarrow p_u + \gamma(e_{ui}q_i - \lambda p_u) \quad (4)$$

donde γ representa la tasa de aprendizaje y λ la extensión de la regularización, los cuales son parámetros que se suelen obtener a través de una técnica llamada *cross-validation*, que consiste en utilizar distintas combinaciones del set de datos iniciales para entrenar y probar el modelo, de forma en que se obtenga un parámetro que ajuste mejor el modelo a los datos.

Existen casos en que un usuario suele valorar los objetos con mayor o menor puntuación que la media, o casos en que un objeto se vuelve popular y recibe una mejor valoración que el resto de los objetos similares a este. Para reflejar este comportamiento en la predicción sea agrega un bias por objeto y por usuario, de forma que la predicción se verá reflejada en la siguiente expresión:

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T p_u \quad (5)$$

donde μ representa el promedio de todas las valoraciones, b_i representa el bias del objeto i y b_u representa el bias del usuario u . La valoración hecha por un usuario u hacia un objeto i se representa con r_{ui} , por otra parte, la predicción de la valoración del usuario u hacia un objeto i se representa con $\hat{r}_{ui}$. Para que el modelo pueda aprender los parámetros b_i , b_u , q_i y p_u , se debe minimizar el error cuadrático en el set de valoraciones conocidas:

$$\min \sum_{(u,i) \in \mathcal{R}} (r_{ui} - \mu + b_i + b_u + q_i^T p_u)^2 + \lambda(b_i^2 + b_u^2 + \|q_i\|^2 + \|p_u\|^2) \quad (6)$$

ALS

Alternating Least Squares es un algoritmo de factorización matricial que consiste en calcular alternadamente los valores de las matrices de usuarios y de objetos, donde al fijar los valores de una de las matrices, el problema se vuelve en una búsqueda de mínimos cuadrados, lo que resulta en una optimización del tiempo de resolución del problema. Se inicializan aleatoriamente los valores de una de las matrices y se calculan los valores de la matriz restante, con el objetivo de minimizar la función de costos, y posteriormente, se recalculan los valores de la matriz cuyos valores fueron previamente fijados. Este proceso se repite hasta que el resultado converja o hasta que se cumpla el máximo de iteraciones. Este método es útil para sistemas recomendadores basados en *feedback* implícito, ya que es capaz de manejar de manera óptima los datos desconocidos, ya que al ser una matriz *sparse*, no debe iterar sobre todos los valores como lo hace el gradiente descendiente en los métodos basados en SVD.

[Hu *et al.*, 2008] adaptó la implementación de ALS propuesta por [Bell y Koren, 2007], donde en lugar de utilizar una función de costo basada en *feedback* explícito, se ajusta para poder recibir *feedback* implícito. Se define el concepto de confianza, el cual a diferencia de los ratings que miden que tanto le gusta un objeto a un usuario, mide que tanta seguridad se tiene de que a una persona le puede llegar a interesar un objeto, dada la cantidad de interacciones se tiene con el objeto, donde r_{ui} representarán las interacciones del usuario u con el objeto i y p_{ui} se denominará como preferencias e indicará si el usuario u ha interactuado con el objeto i .

$$p_{ui} = \begin{cases} 1 & r_{ui} > 0 \\ 0 & r_{ui} = 0 \end{cases} \quad (7)$$

La confianza se ve denominada por la siguiente expresión:

$$c_{ui} = 1 + \alpha r_{ui} \quad (8)$$

Donde α regulará el incremento de la confianza. La función de costos se verá representada de la siguiente forma:

$$\min_{x,y} \sum_{u,i} c_{ui} (p_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2 \right) \quad (9)$$

Donde x_u representará el vector de factores latentes del usuario u , y_i representará el vector de factores latentes del objeto i y λ será un parámetro que regularizará el sobreajuste del modelo, el cual se obtiene por *cross-validation*.

Para recalcular la matriz de usuarios, se debe obtener cada vector de usuarios de manera independiente. se denominará como x_u como el vector de factores latentes para el usuario u , Y será la matriz de productos, C^u será una matriz diagonal, donde C_{ii}^u contendrá la confianza del usuario u por el objeto i y $p(u)$ contendrá las preferencias del usuario u . Así, la expresión para obtener x_u de forma que minimice la función de costos es la siguiente:

$$x_u = (Y^T C^u Y + \lambda I)^{-1} Y^T C^u p(u) \quad (10)$$

Una vez obtenida esta matriz, se podrá a recalcular la matriz de productos. De manera similar al caso de la obtención de la matriz de usuarios, se denominará como y_i al vector de factores latentes del objeto i , X representará a la matriz de usuarios obtenida previamente, C^i será una matriz diagonal, donde C_{uu}^i contendrá la confianza del usuario u por el objeto i y $p(u)$ contendrá las preferencias hacia el objeto i .

$$y_i = (X^T C^i X + \lambda I)^{-1} X^T C^i p(i) \quad (11)$$

2.2. ANÁLISIS DE CANASTA

El análisis de canasta es una técnica utilizada por las grandes empresas de *retail* para descubrir asociaciones entre productos [Li, 2017], mediante la observación de las combinaciones de aquellos que aparecen con mayor frecuencia en transacciones. Para lograr esto, se establecen reglas de asociación, las cuales ponen un límite sobre distintas métricas de frecuencia de aparición de productos, ya sea que aparezcan de forma dependiente o independiente.

Al estudiar la dependencia entre productos, estos se clasifican de la siguiente forma:

- **Antecedente:** Producto que influye en la compra de otro producto.
- **Consecuente:** Producto que se compra dado que se compró el antecedente.

Para establecer las reglas de asociación, se utilizan las siguientes métricas:

- **Soporte:** Frecuencia con la que se compra un producto dado el total de boletas.
- **Soporte Conjunto:** Frecuencia con la que dos o más productos son comprados juntos dado el total de boletas.
- **Confianza:** Frecuencia con la que dos o más productos son comprados juntos dado el total de boletas que incluyen al antecedente.
- **Lift:** Métrica que indica si dos objetos son independientes entre sí (Valor menor o igual a 1) o si son complementarios (Valor mayor a 1).

2.3. METODOLOGÍAS PARA PROYECTOS DE ANÁLISIS DE DATOS

Con el pasar de los años, los procesos de minería de datos se han vuelto altamente implementados en las industrias, lo que provocó la necesidad de identificar buenas prácticas para evitar cometer errores a partir de experiencias previas. A continuación, se abordarán las metodologías CRISP-DM, KDD, SEMMA y DST.

2.3.1. CRISP-DM

CRISP-DM (*Cross-Industry Standard Process for Data Mining*) es una metodología de trabajo enfocada en proyectos de minería de datos [Chapman *et al.*, 2000]. Es un proceso flexible de 6 pasos, que permite volver a una fase previa cuando el resultado de la fase actual no sea el esperado. En la figura 2 se puede apreciar el flujo de la metodología.

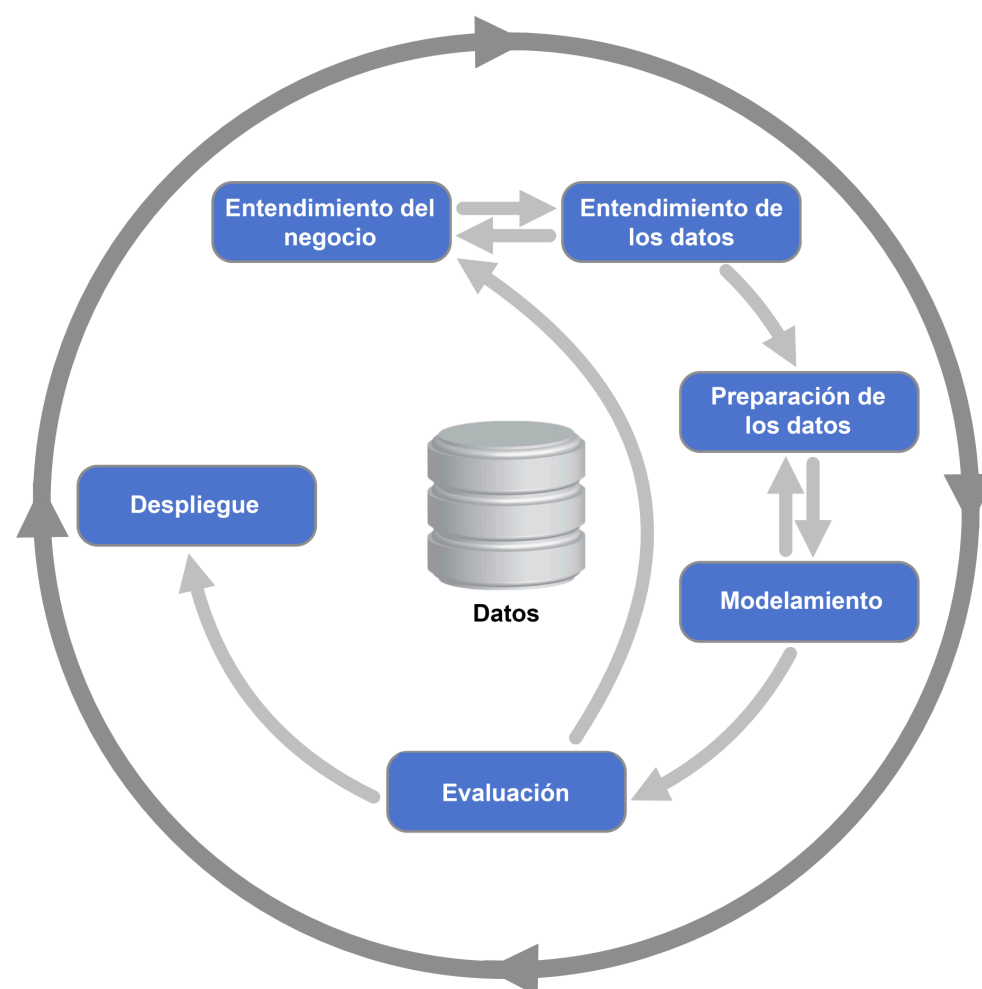


Figura 2: Fases de la metodología CRISP-DM.
Fuente: [Jensen, 2012]

Las fases de CRISP-DM son las siguientes:

- **Entendimiento del negocio:** Es el paso inicial de CRISP-DM. Consiste en entender las necesidades del negocio para poder transformar el conocimiento obtenido en un problema de minería de datos y poder plantear los objetivos a resolver.
- **Entendimiento de los datos:** En esta fase se estudian los datos para poder familiarizarse con ellos, de forma que se logre detectar problemas en la calidad de los datos, descubrir relaciones entre entidades y formular hipótesis a partir de ellos.
- **Preparación de los datos:** Esta fase abarca todas las actividades necesarias para poder obtener el set de datos final para posteriormente realizar el modelamiento de datos. Estas actividades incluyen limpieza, transformación y selección de los datos.
- **Modelamiento:** En esta fase se aplican técnicas de modelamiento de datos. Al aplicar distintas técnicas, puede ser necesario volver a la fase Data Preparation para poder llevar los datos al formato requerido.
- **Evaluación:** En esta fase se evalúa si los modelos obtenidos en la fase anterior satisfacen los objetivos propuestos. Se debe revisar si alguna necesidad del negocio no fue considerada en la construcción de los modelos.

- **Despliegue:** En esta fase se presenta la información obtenida a partir de los modelos. La forma en que se presenta la información puede variar según las necesidades del cliente, el cual puede requerirla a través de un reporte o puede buscar obtener resultados actualizados en tiempo real.

2.3.2. KDD

KDD (*Knowledge Discovery in Databases*) es una metodología para la extracción automatizada del conocimiento en bases de datos, donde la minería de datos no es el foco principal sino un paso del proceso [Fayyad *et al.*, 1996]. En la figura 3 se puede apreciar el flujo de la metodología.

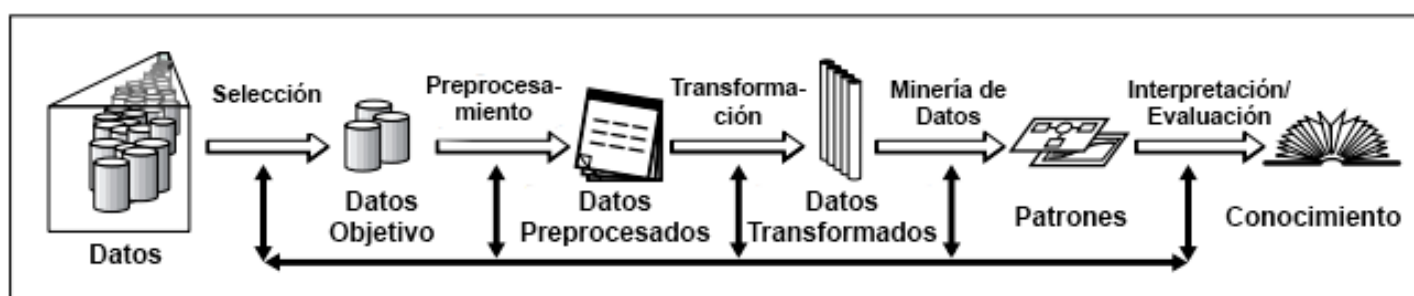


Figura 3: Fases de la metodología KDD.

Fuente: [Fayyad *et al.*, 1996]

Las fases de KDD son las siguientes:

- **Selección:** El primer paso consiste en construir el set de datos inicial, donde se buscará descubrir información. Para construir el set es posible seleccionar las variables necesarias u obtener una muestra de los datos.
- **Preprocesamiento:** En esta fase se realiza la limpieza de datos, en la cual se eliminan los *outliers*, se manejan los registros con campos vacíos y se resuelven problemas relacionados al sistema de gestión de bases de datos.
- **Transformación:** En esta fase se busca descubrir características útiles para representar a los datos. También se busca reducir el número de variables a considerar o encontrar representaciones invariantes para los datos.
- **Minería de Datos:** En esta fase se realiza el proceso de minería de datos, que consta de encontrar el objetivo del modelo, escoger el algoritmo apropiado para encontrar patrones en los datos, identificar los parámetros adecuados y encontrar una representación para los patrones identificados.
- **Interpretación/Evaluación:** En esta fase se interpretan y visualizan los patrones obtenidos, donde se eliminan los resultados redundantes e irrelevantes. El conocimiento

obtenido puede ser documentado, incorporado en la toma de decisiones o implementado en el rendimiento de un sistema.

2.3.3. SEMMA

SEMMA (*Sample, Explore, Modify, Model, Assess*) es una serie de pasos secuenciales desarrollada por SAS Institute para su herramienta SAS Enterprise Miner [SEMMA, 2012]. En la figura 4 se puede observar el flujo de esta metodología.

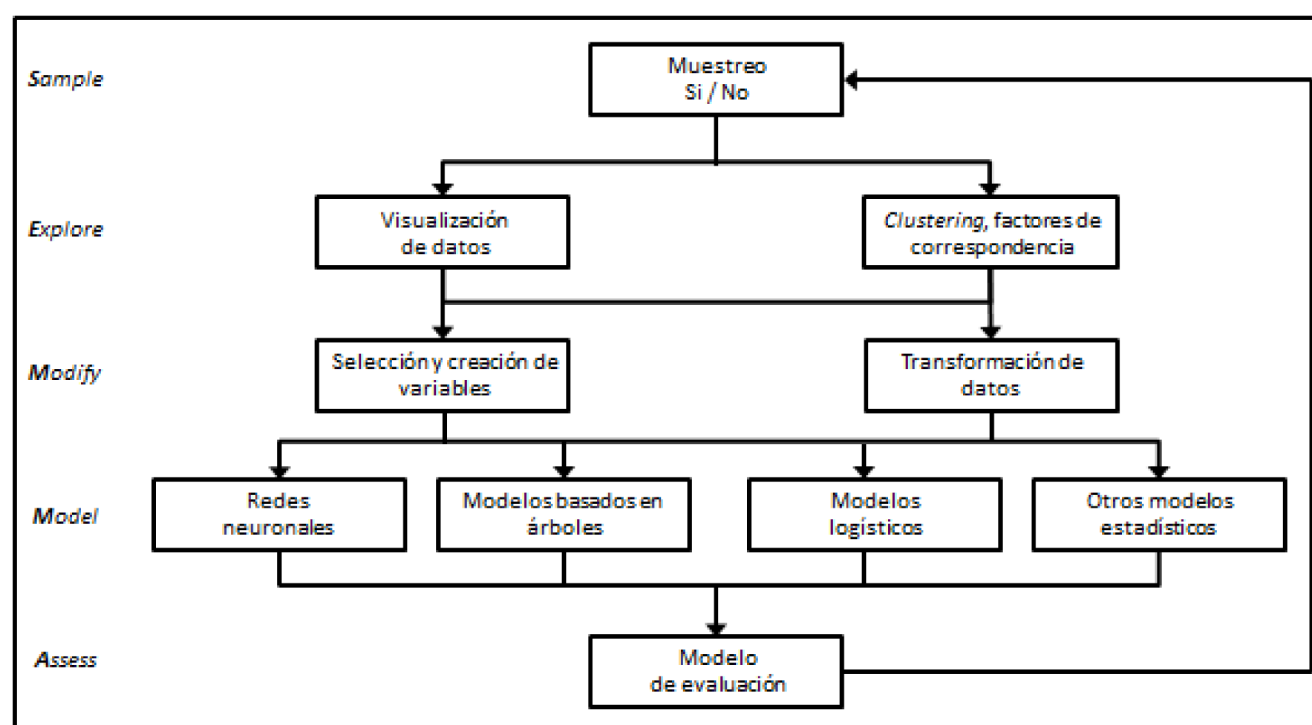


Figura 4: Fases de la metodología SEMMA.

Fuente: [Peralta, 2014]

Las fases de SEMMA son las siguientes:

- **Muestreo:** Esta fase es opcional, la cual consiste en obtener una muestra de datos. Esta muestra debe ser lo suficientemente grande como para contener información relevante, pero lo suficientemente pequeña para poder manipularla de manera eficiente.
- **Exploración:** En esta fase se explora el set de datos para encontrar tendencias y anomalías, con el fin de generar conocimiento e ideas. Si una exploración visual no logra revelar tendencias, es posible usar técnicas estadísticas en esta fase.
- **Modificación:** Esta fase consiste en modificar el set de datos para poder seleccionar un modelo. Esta modificación consiste en crear, seleccionar y transformar las variables de los datos. En este proceso se debe decidir cómo manejar los outliers. También se debe reducir el número de variables para quedarse con las más significativas.

- **Modelado:** En esta fase se aplican técnicas de modelado sobre el set de datos, de forma que se logren predecir resultados confiables y precisos. Algunas de estas técnicas pueden ser redes neuronales artificiales, árboles de clasificación, regresiones logísticas u otros modelos estadísticos.
- **Evaluación:** En esta fase se evalúan los modelos para verificar que los descubrimientos sean útiles y confiables. Al terminar esta fase pueden surgir nuevas preguntas, por lo que se puede volver a la fase de exploración para obtener datos más refinados.

2.3.4. DST

DST (*Data Science Trajectories*) es una metodología de trabajo que surgió a partir de la metodología CRISP-DM. En los últimos 20 años, se ha pasado a hablar de minería de datos a ciencia de datos. Esta última abarca una gran variedad de proyectos, los cuales no se centran solamente en la exploración, sino que cuentan con fases enfocadas al cumplimiento de objetivos. Como CRISP-DM surgió como una metodología enfocada a la minería de datos, los proyectos de ciencia de datos pueden verse estancados al ser desarrollados mediante esta metodología, ya que pueden verse cortados al terminar en un paso intermedio o pueden carecer de fases de exploración intermedias. DST propone una metodología sin trayectorias predefinidas, en donde el líder del proyecto es el encargado de definir el orden en base a la experiencia de proyectos anteriores [Martinez-Plumed *et al.*, 2021]. La metodología DST propone una variedad de actividades, donde incluye actividades de exploración y gestión de datos, además de las actividades propuestas por la metodología CRISP-DM. En la figura 5 se puede observar que las actividades externas corresponden al proceso de exploración, las intermedias corresponden a las actividades orientadas a objetivos de CRISP-DM y las interiores a la gestión de datos.



Figura 5: Mapa con las actividades de DST.
Fuente: [Martinez-Plumed *et al.*, 2021]

Las actividades de exploración son las siguientes:

- **Exploración de objetivos:** Se buscan los objetivos del negocio, los cuales pueden ser logrados mediante soluciones basadas en datos.
- **Exploración de fuentes de datos:** Se descubren fuentes de datos nuevas y valiosas.
- **Exploración del valor de los datos:** Se descubre el valor que puede ser extraído de los datos.
- **Exploración de resultados:** Se busca relacionar los resultados obtenidos mediante técnicas de ciencia de datos a los objetivos del negocio.
- **Exploración de la narración:** Se extraen historias a partir de los datos, las que pueden ser textuales o visuales.
- **Exploración del producto:** Se busca la forma de llevar la información obtenida de los datos a un servicio o aplicación que le entregue valor al cliente.

Las actividades de gestión de datos son las siguientes:

- **Adquisición de datos:** Se crean o se obtienen datos relevantes a través de distintos medios como aplicaciones o sensores.
- **Simulación de datos:** Se simulan sistemas complejos para poder producir datos útiles.

- **Arquitectura de datos:** Se diseña el modelo lógico y físico de los datos, además de integrar diferentes fuentes de datos.
- **Liberación de datos:** Los datos se dejan disponibles para su uso mediante bases de datos, interfaces y visualizaciones.

CAPÍTULO 3

PROPUESTA DE SOLUCIÓN

En el presente capítulo se explica la solución propuesta para enfrentar el problema presentado en el capítulo 1. Esta se basa en implementar un sistema recomendador de filtrado colaborativo para poder predecir qué productos serán del agrado de cada usuario, con el fin de brindar una experiencia personalizada en el *E-Commerce* de Muevo. Esto se realizará modelando el *feedback* implícito obtenido desde la aplicación, el cual corresponde a las transacciones hechas por los usuarios en las tiendas Pronto. Para el desarrollo de este proyecto, es necesario seleccionar una de las metodologías de trabajo presentadas en el capítulo 2, de forma que se escoja aquella que se adapte mejor a las necesidades del proyecto.

3.1. Metodología de trabajo

Para el desarrollo de este trabajo, se decidió aplicar la metodología CRISP-DM, debido a que, a diferencia de KDD y SEMMA, considera al negocio como parte fundamental del proceso, el cual está presente en la fase inicial de entendimiento y en la fase final de evaluación, por lo que se adapta bien a las necesidades del proyecto. Por otra parte, se descartó la metodología DST, ya que se optó por ultizar una metodología más simplificada dada la extensión del proyecto, además de que los pasos planteados por CRISP-DM son suficientes para abarcar las tareas a desarrollar.

En la tabla 1 se presentan las etapas del proyecto y se describirá brevemente cada una de ellas, para posteriormente profundizarlas en las siguientes secciones.

Tabla 1: Fases de CRISP-DM.

Fuente: Elaboración Propia.

Fase	Descripción
Entendimiento del negocio	Se tienen reuniones con el negocio para entender las necesidades del equipo de trabajo de Muevo
Entendimiento de los datos	Se realizan análisis para entender el comportamiento de los datos transaccionales
Preparación de los datos	Se realizan procesos de extracción y transformación de los datos transaccionales para entrenar al modelo
Modelamiento	Se aplican técnicas de modelamiento para generar la matriz de predicciones
Evaluación	Se evalúan los resultados obtenidos mediante experimentos reales y métricas de desempeño
Despliegue	Se proponen casos de uso para aplicar el sistema recomendador a Muevo

3.2. Herramientas utilizadas

Para el desarrollo del sistema recomendador se utilizó el lenguaje de programación *Python* a través de la aplicación *Jupyter Notebook*. Además, se hizo uso de la plataforma *Amazon SageMaker* para optimizar el tiempo de entrenamiento de los sistemas recomendadores.

Python

Python es una lenguaje de programación de alto nivel cuya característica es la legibilidad de su código [pyt, 1991]. Se considera multiparadigma ya que soporta parcialmente orientación a objetos, programación imperativa y programación funcional. Es altamente utilizado para realizar análisis de datos y desarrollar proyectos de ciencia de datos.

Jupyter Notebook

Jupyter Notebook es una aplicación web interactiva que permite crear documentos [jup, 2014], los cuales se almacenan en formato JSON y contienen una lista ordenada de celdas que pueden almacenar código, texto o gráficos. El nombre *Jupyter* es un acrónimo de los lenguajes *Julia*, *Python* y *R*, sin embargo soporta múltiples lenguajes de programación.

Amazon SageMaker

Amazon SageMaker es una plataforma de aprendizaje automático en la nube lanzada por *Amazon Web Services* [sag, 2017]. Esta plataforma permite crear, entrenar e implementar modelos de aprendizaje automático. Provee una infraestructura optimizada para reducir los tiempos de entrenamiento.

3.3. Desarrollo del Proyecto

En esta sección se plantea cómo se aborda cada uno de los pasos de la metodología CRISP-DM, para posteriormente en el capítulo 4 explicar el desarrollo de cada uno de ellos para la construcción de la solución propuesta.

3.3.1. Entendimiento del negocio

Antes de comenzar un proyecto de ciencia en datos, es necesario entender las necesidades del negocio, ya que el modelo a desarrollar deberá dar solución a una problemática en la empresa. Una mala comprensión del problema puede llegar a generar un modelo que no entregue valor al negocio, lo cual implicaría un desperdicio de recursos.

Para poder entender las necesidades de Copec y lograr familiarizarse con la aplicación Muevo, se tuvieron reuniones con los ingenieros de proyectos que están a cargo de las distintas secciones de la aplicación. En estas reuniones, cada uno de los ingenieros presentaron los proyectos en los cuales están trabajando actualmente, los objetivos que buscan alcanzar y los KPIs que quieren llegar a impactar, tomando como punto de inicio abril del 2022 y con fecha límite diciembre del mismo año. Para el desarrollo de este proyecto, se trabajó solamente con la ingeniería a cargo del E-Commerce de Muevo.

La ingeniera de proyectos del E-commerce de Muevo (Que desde ahora en adelante llamaremos *stakeholder*) señaló que, para la fecha, Muevo es una aplicación estática, cuya interfaz es igual para todas las personas que ingresen a ella, por lo que independiente de los gustos y preferencias que tengan los usuarios, a todos se les va a mostrar los productos en el mismo orden, ya sea en la página de inicio, en el catálogo de Full, y en el catálogo de Escanea y Paga, donde puede darse la situación en que los primeros productos en aparecer no lleguen a ser del agrado de los usuarios, y por ende tomen la decisión de no adquirir los productos o promociones ofrecidas, incluso no canjear premios debido a que a simple vista no parecen ser atractivos para ellos. En la figura 6 se puede ver la página de inicio de Muevo, donde se puede ver el carrusel de promociones que se pueden comprar en tiendas Pronto, donde el orden de aparición de las promociones es estático. En la figura 7 se muestra el catálogo de Escanea y Paga, donde se puede ver que la primera categoría es la de Pizzas, la cual puede no ser atractiva a primera vista para algunos usuarios, como aquellos que prefieren comida más saludable.

Es posible que el usuario, al no sentirse a gusto con los primeros productos en ser presentados, decida no buscar dentro del catálogo ofrecido por la aplicación, perdiendo la oportunidad de encontrar algún producto que sea de su interés y, por ende, perdiendo la posibilidad de aumentar el ticket promedio y el promedio de productos por canasta.

El equipo de *insights* de Copec realizó una investigación cualitativa al ciclo de vida de los clientes de Copec, donde se detectó que para algunos clientes, el canje de cupones es un hito que influye en la decisión de cargar combustible. El usuario, al canjear el cupón y tangibiliza el beneficio, se siente recompensado por Copec y fortalece su vínculo con la marca. Sin embargo, muchos usuarios no canjean los cupones ya que consideran que los beneficios son poco atractivos. Es por eso que el equipo de *insights* propone fomentar el canje de cupones a través de recomendaciones personalizadas.



Figura 6: Página de inicio de Muevo.
Fuente: Elaboración Propia.

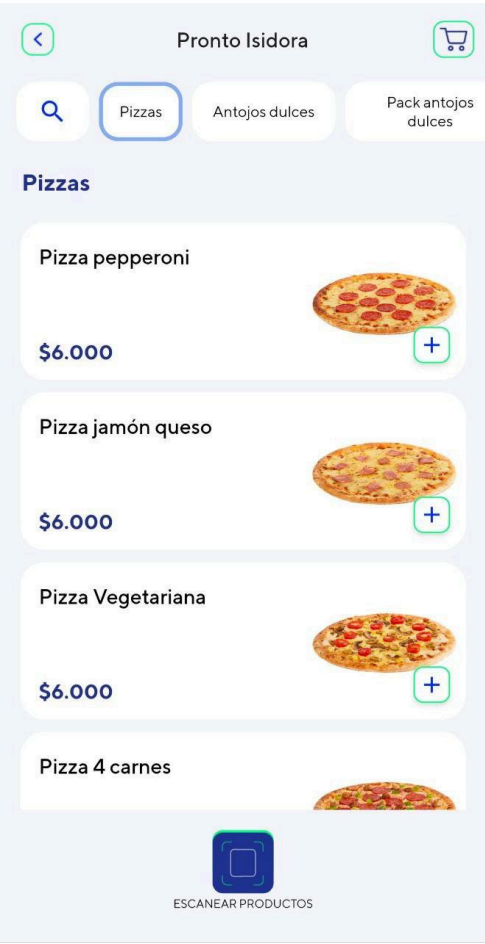


Figura 7: Página de inicio de Escanea y Paga.
Fuente: Elaboración Propia.

Por los motivos mencionados previamente es que a la *stakeholder* le interesa que el *E-Commerce* de Muevo llegue a brindar una experiencia personalizada, donde a cada usuario se le muestren productos que potencialmente puedan llegar a ser de su interés, con el objetivo de que se sienta más a gusto al utilizar la aplicación, de forma en que se vuelva leal a la marca y tiendan a comprar los productos que se le recomiendan.

3.3.2. Entendimiento de los datos

Para poder construir el sistema recomendador, se debe contar con los datos transaccionales de los clientes de las tiendas Pronto, por lo que es necesario que el gobierno de datos de Copec provea los accesos correspondientes para la extracción de los datos.

Además de contar con el acceso a los datos, se debe entender qué valor se puede inferir a través de ellos, por lo que se lleva a cabo un análisis exploratorio de datos, en el cual se busca conocer cuáles son los productos más comprados, entender si existen productos con una mayor tendencia a ser comprados juntos, y descubrir si el contexto en el que se realiza una transacción influye en las canastas de compra.

Extracción de datos

Para la obtención inicial de datos, se habilitó una cuenta en el *data lake* de Copec. En la tabla 2 se señalan cuales son las bases de datos a las cuales se otorgó acceso junto a la información que almacenan:

Tabla 2: Bases de datos de Copec.

Fuente: Elaboración Propia.

Base de datos	Descripción
copec_fidelidad_prd	Almacena principalmente información de las transacciones realizadas en Copec, ya sea en Estaciones de Servicio, Tiendas Punto o Pronto. También almacenan información de cupones, promociones y usuarios.
copec_maestros_analytics_prd	Almacena información de tiendas, estaciones de servicio, productos, clientes y surtidores de combustible.
copec_meteored_prd	Almacena el reporte del tiempo para locaciones selectas de Chile. Incluye datos como la fecha de medición, la descripción del tiempo, temperatura mínima y máxima.

En la tabla 3 se indican cuales fueron las tablas que se utilizaron para realizar el análisis explo-

ratorio de datos, la base de datos a las que pertenecen y cuales son los datos que almacenan:

Tabla 3: Tablas utilizadas para el EDA.

Fuente: Elaboración Propia.

Tabla	Base de Datos	Descripción
trx_cabecera_rl	copec_fidelidad_prd	Almacena información sobre en donde, cuando y quien emitió la boleta
trx_detalle_rl	copec_fidelidad_prd	Almacena información de los productos que se compraron en las boletas
tb_maestro_producto_rl	copec_maestros_analytics_prd	Almacena el nombre, el código y las categorías de todos los productos
tb_maestro_tiendas_rl	copec_maestros_analytics_prd	Almacena los código, la locación, el formato, el encargado y la estación de servicio asociada de todas las tiendas
pronostico_rl	copec_meteored_prd	Almacena la descripción del tiempo, la fecha de medición, la temperatura mínima y máxima y los milímetros de lluvia para 49 locaciones de Chile

Al empezar a trabajar sobre estas tablas, se detectaron ciertos problemas. En la tabla maestro de productos existe un total de 1121 códigos que tienen asociado más de un producto, y 1488 productos que tienen más de un código asociado. Por otra parte, 2306 códigos de productos que aparecen asociados a boletas en la tabla detalle no pudieron ser encontrados en el maestro de productos. Para evitar este problema, se decidió no utilizar los nombres de grupos y de productos de la tabla maestro de productos, sino que se utilizaron los que aparecen en la tabla detalle; sin embargo, los grupos de productos de esa tabla no están estandarizados, ya que existen muchas categorías que corresponden a un solo producto o aparecen en muy pocas tiendas. Para trabajar con datos más limpios, se optó por trabajar con aquellas grupos que aparezcan en 10 o más tiendas y que tengan un alcance en transacciones mayor al 0.1 %, además de seleccionar aquellas transacciones que hayan sido hechas en tiendas que trabajen con el sistema Sckuba, ya que los grupos que aparecen en estas transacciones abarcan más de un producto en específico.

Dados los problemas mencionados previamente, la stakeholder proveyó dos tablas en formato .xlsx, las cuales corresponden a un maestro de tiendas y uno de productos. A diferencia de la información que se puede extraer de las tablas de copec_maestros_analytics_prd, los

nuevos maestros contienen datos actualizados de tiendas y productos. En el caso del maestro de tiendas, este indica el sistema con el cual trabaja la tienda, por lo cual podemos filtrar aquellas tiendas que pertenezcan al sistema Sckuba. Por otra parte, en el caso del maestro de productos, los códigos de productos y los nombres de categorías están mejor catalogados para tiendas que pertenezcan al sistema Sckuba, donde se señala que aunque el nombre del producto llegue a cambiar, el código siempre indicará al mismo producto.

Análisis Exploratorio de Datos

Para entender los patrones de consumo de los clientes en tiendas Pronto, se hizo una análisis exploratorio de datos, en el cual se buscó conocer cuales son los productos más consumidos y descubrir si hay productos que poseían una tendencia a ser comprados juntos mediante un análisis de canasta.

Se realizó el análisis sobre 3.223.993 transacciones únicas, las cuales se realizaron desde el 1 de Enero del 2022 hasta el 31 de Julio de 2022 en un total de 28 tiendas Pronto. Se consideraron estas tiendas ya que son las únicas que actualmente trabajan con el sistema llamado Sckuba. Estas transacciones consideran 2418 productos, los cuales se clasifican en 21 categorías diferentes.

En la figura 8 se pueden apreciar los 20 productos con mayor aparición en boletas. Entre ellos destacan la Coca Cola, hamburguesas, hot dogs, mechadas y Red Bull, lo cual se podría asociar a almuerzos o consumos en carretera. Por otra parte, con menor aparición, se puede encontrar productos asociados a desayunos y onces como lo son los té de variedades, combos desayuno de huevo, hot dog italiano, vital pet 600c s/g, capuccino (m), porcion empanaditas, cachmentun 500 cc s/g, vital pet 600c c/g, red bull sugar free, miga ave pimenton fresco.

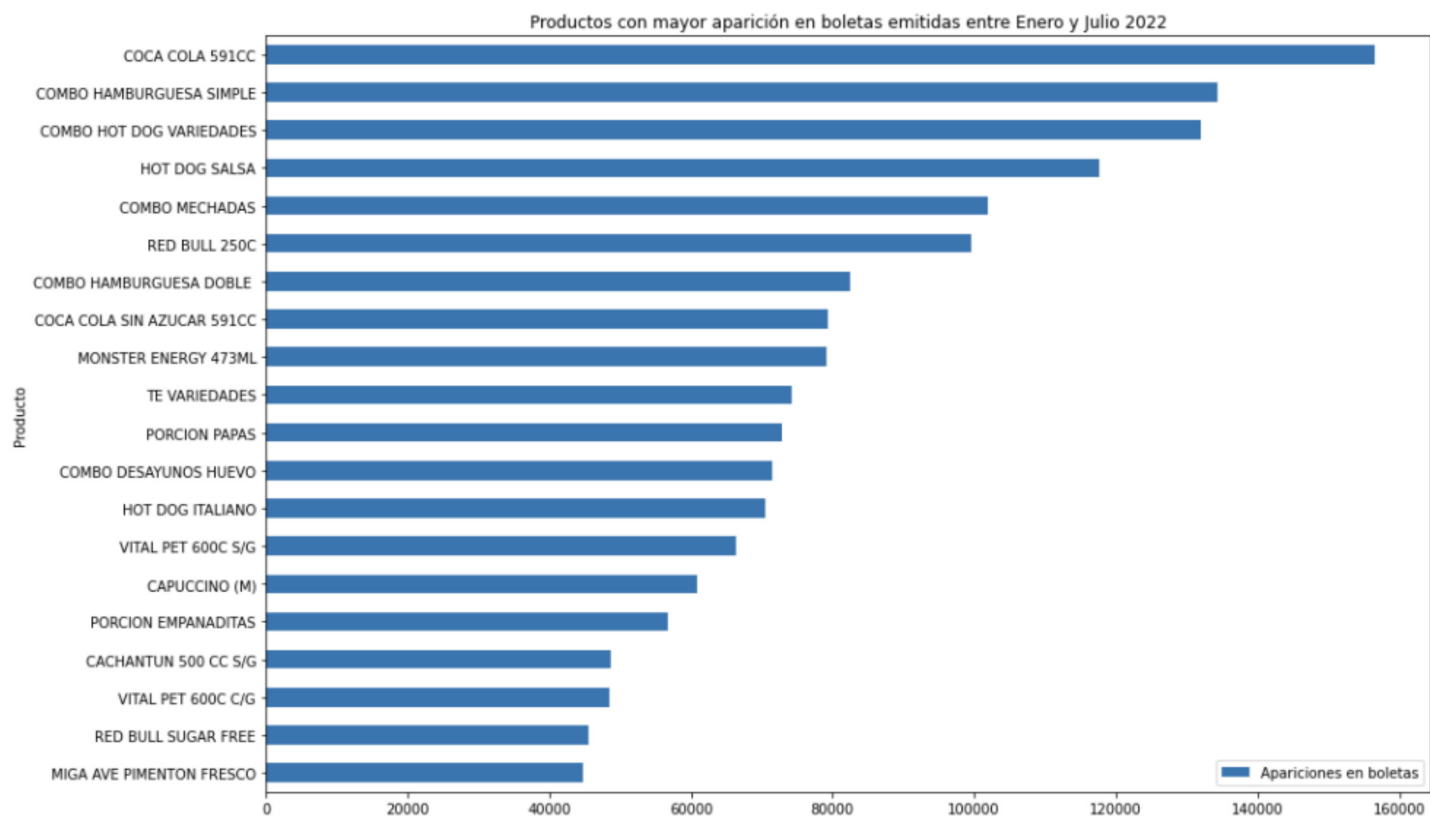


Figura 8: Productos con mayor aparición en boletas.
 Fuente: Elaboración Propia.

En la figura 9 se pueden ver los 20 grupos de productos con mayor aparición en boletas, donde el grupo que aparece con mayor frecuencia es Carro Isla, el cual representa a aquellos productos que se ubican en un estante denominado isla dentro de las tiendas Pronto, por lo cual los productos en este grupo tienden a cambiar. Seguido de esto se puede ver que destacan los grupos Cafetería y Confitería, y posterior a ellos se pueden encontrar los grupos Sandwich y Pizzas & Hot Dogs, lo cual no parece hacer sentido con el orden en que aparecen los productos en la figura 8. Esto se puede deber a que algunos productos, además de estar en su respectivo grupo, también se pueden encontrar en el grupo Carro Isla, como lo es el caso de la Coca Cola y la Red Bull, lo que provoca que grupos como Bebidas y Energéticas estén bajo los otros grupos a pesar de poseer algunos de los productos más vendidos.

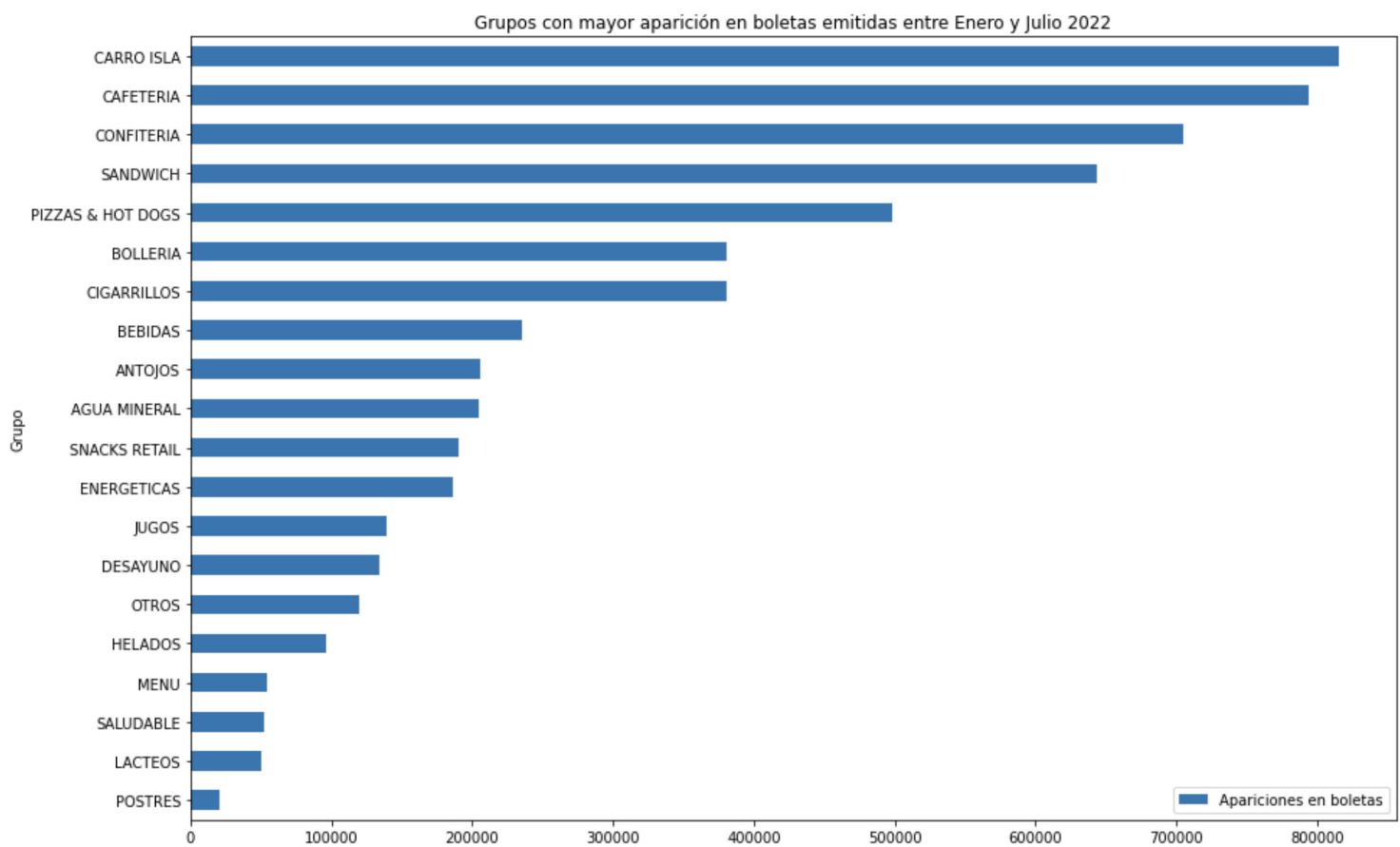


Figura 9: Grupos con mayor aparición en boletas.
Fuente: Elaboración Propia.

La *stakeholder* planteó la idea de que se generaran recomendaciones de productos en base al contexto en el que se sitúe el cliente. Es por eso que se estudió factores como el tiempo, la hora o el tipo de estación influyen en cuales son los productos más comprados y en las canastas de compra. Para eso se definieron los siguientes contextos:

1. Momento del día: Corresponde al momento del día en que se realiza la compra. Se consideraron 4 segmentos de 6 horas cada uno:
 - Mañana: 06:00 am - 11:59 am.
 - Tarde: 12:00 pm - 17:59 pm.
 - Noche: 18:00 pm - 23:59 pm.

- Madrugada: 00:00 am - 05:59 am.
2. Tiempo: Corresponde a los distintas descripciones del tiempo que provee la tabla pronostico_rl, las cuales son las siguientes:
- Lluvias
 - Cielos Cubiertos
 - Cielos Nubosos
 - Intervalos Nubosos
 - Cielos Despejados
3. Formato: Corresponde al tipo de tienda en la que se hace la compra. Se dividen en dos tipos:
- Ciudad: Tiendas Pronto que se ubican dentro de la ciudad.
 - Carretera: Tiendas Pronto que se ubican en carreteras.

A continuación, se visualiza cómo se distribuyen las boletas dados los distintos contextos propuestos, con el objetivo de entender qué tan representativos son estos segmentos y conocer si existe una tendencia.

En la figura 10 se puede ver la distribución de transacciones por momento del día, donde el 35 % de las transacciones se realizaron en la tarde, el 30.7 % de ellas en la noche, el 25.3 % en la mañana y el 9 % en la madrugada.

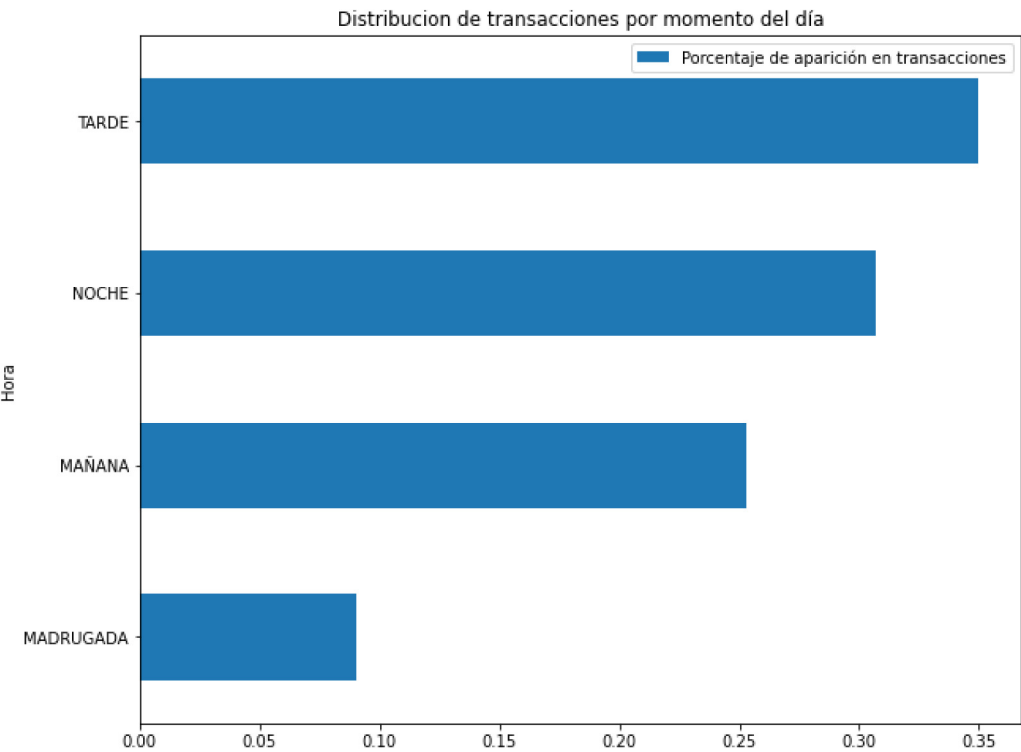


Figura 10: Distribución de transacciones por momento del día.
Fuente: Elaboración Propia.

En la figura 11 se puede ver la distribución las transacciones por tiempo, donde el 34.25 % de las transacciones fueron hechas en Cielos Despejados, el 22.82 % fueron realizadas en Intervalos Nubosos, el 22.17 % en Lluvias, el 20.25 % en Cielos Nubosos y el 0.5 % en Cielos Cubiertos.

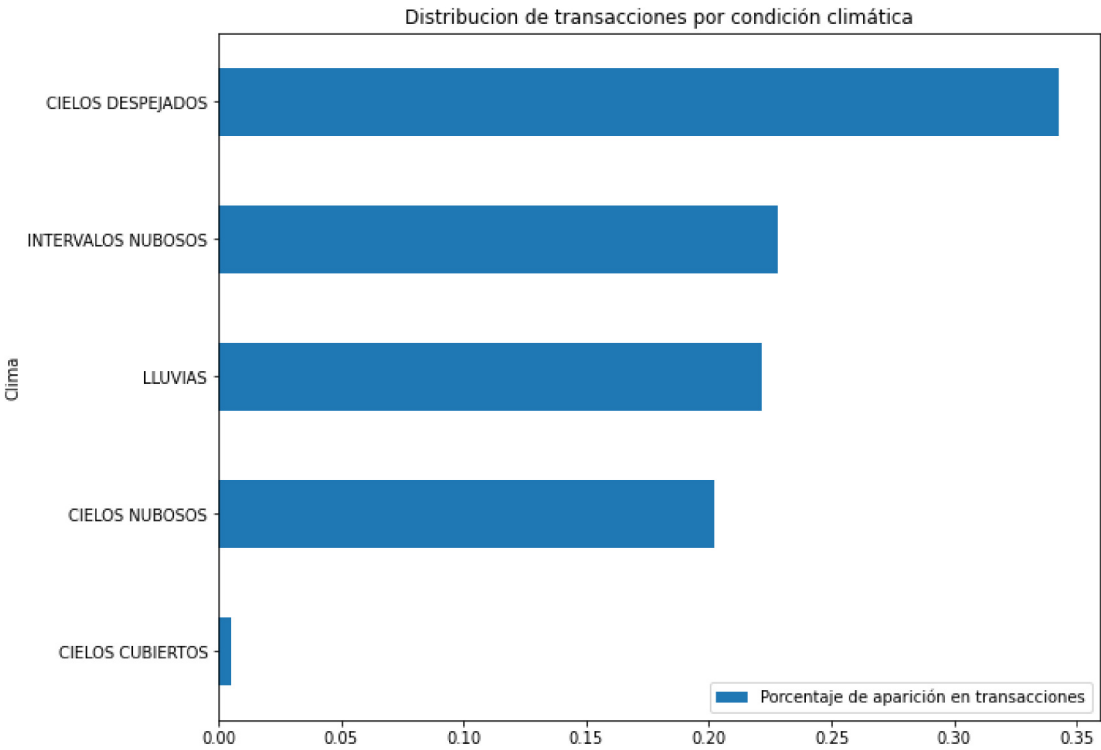


Figura 11: Distribución de transacciones por tiempo.
Fuente: Elaboración Propia.

En la figura 12 se puede ver la distribución de transacciones por formato de tienda, donde el 95.3 % de las transacciones se realizaron en tiendas de carretera y el 4.7 % en tiendas de ciudad, lo cual corresponde a solamente 3 tiendas.

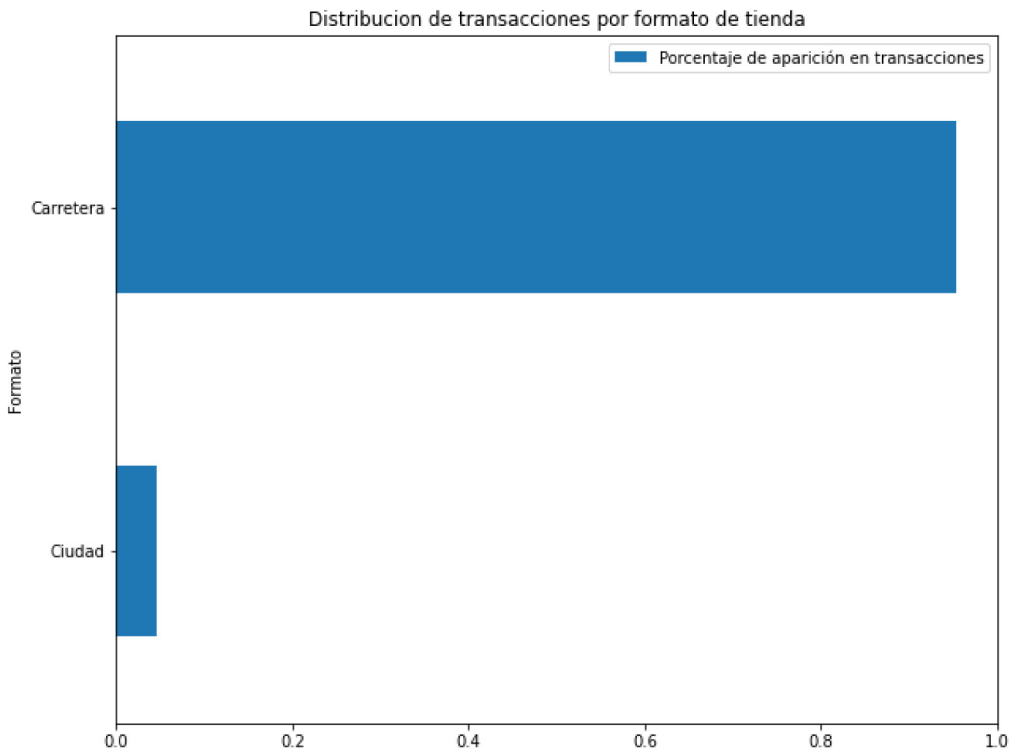


Figura 12: Distribución de transacciones por formato de tienda.
Fuente: Elaboración Propia.

A continuación, se analizan cuáles son los productos con mayor aparición en boletas para cada segmento de cada contexto, con el objetivo de identificar si existen productos populares exclusivos para cada segmento y poder determinar si el contexto es un factor determinante a la hora de decidir que productos comprará un usuario.

En las figuras 13, 14, 15 y 16 se visualizan los 20 productos con mayor aparición en boletas para horarios de mañana, tarde, noche y madrugada respectivamente. Se puede apreciar que varios de esos productos se repiten con los productos de mayor aparición a nivel general (véase la figura 8), con la diferencia de que en la mañana aparecen combos de tapaditos, sandwich miga Barros Jarpa y variedades de cafés; para la tarde, menús de almuerzo, porciones de papas y una Cachantun; en la noche, combos de donuts, hot dogs y papas; y por último en la madrugada, nuevamente las porciones de papas y distintos tipos de cafés, de manera similar a los productos del horario de mañana.

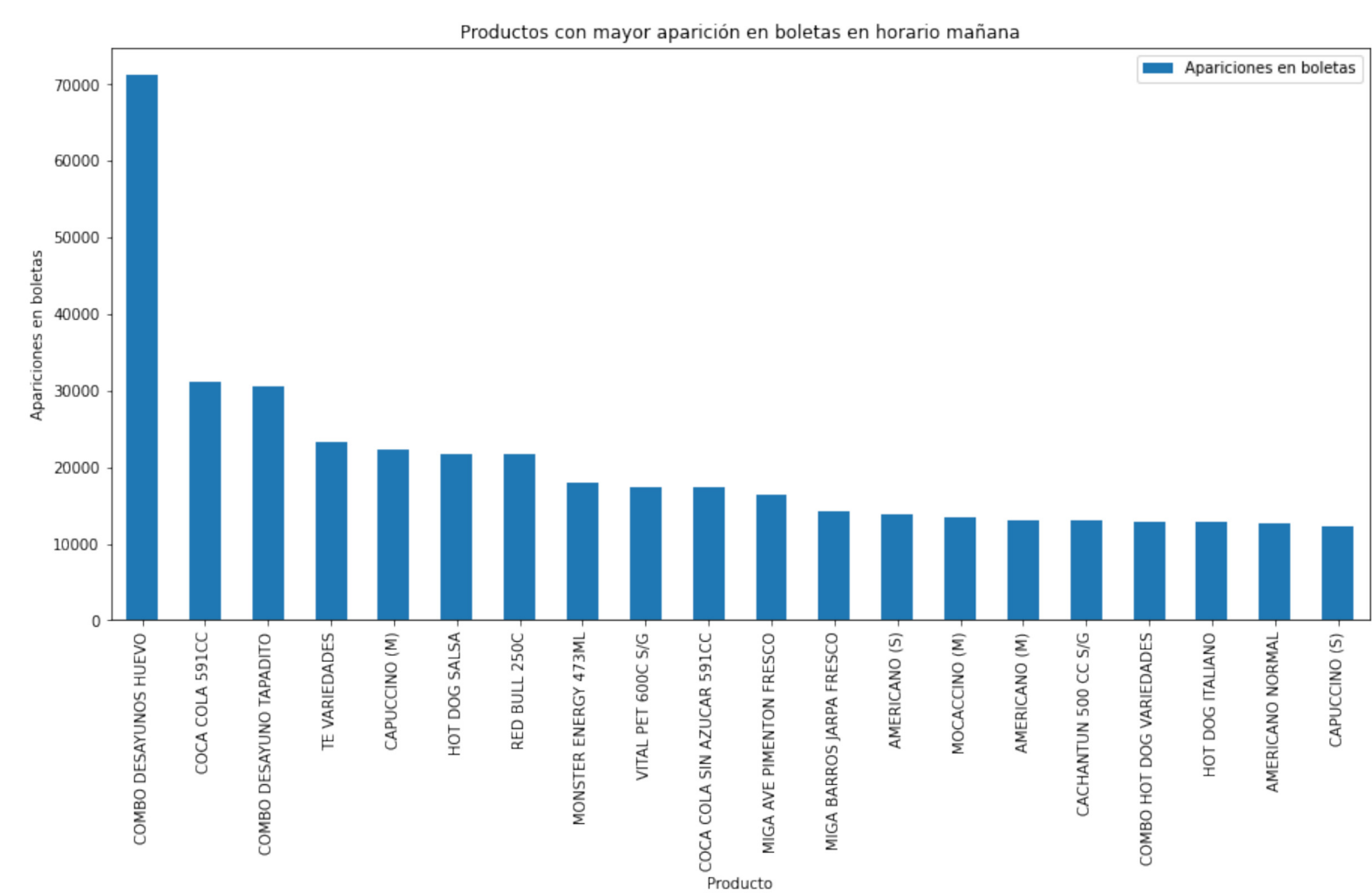


Figura 13: Productos con mayor aparición en boletas en horario de mañana.
Fuente: Elaboración Propia.

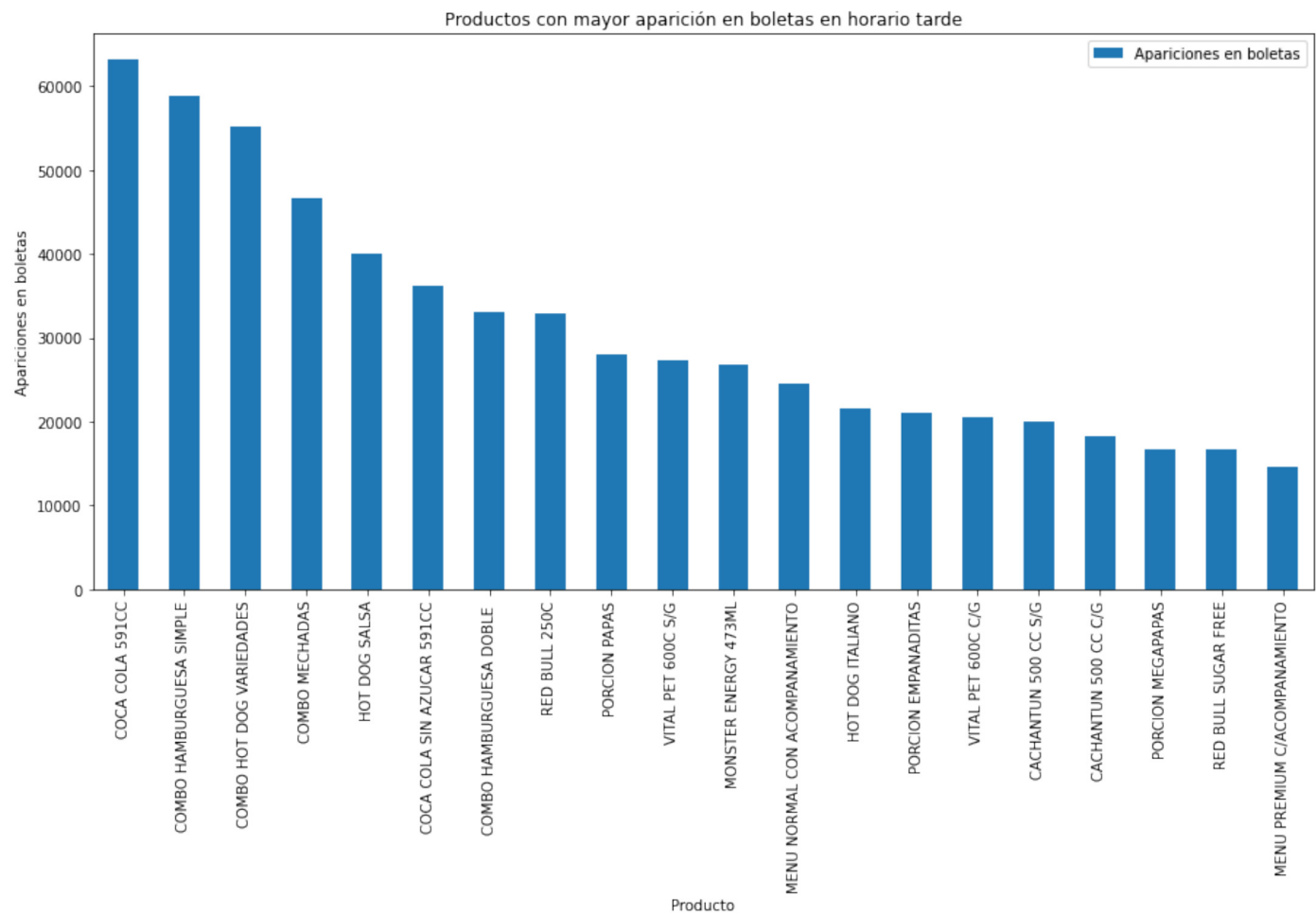


Figura 14: Productos con mayor aparición en boletas en horario tarde.
Fuente: Elaboración Propia.

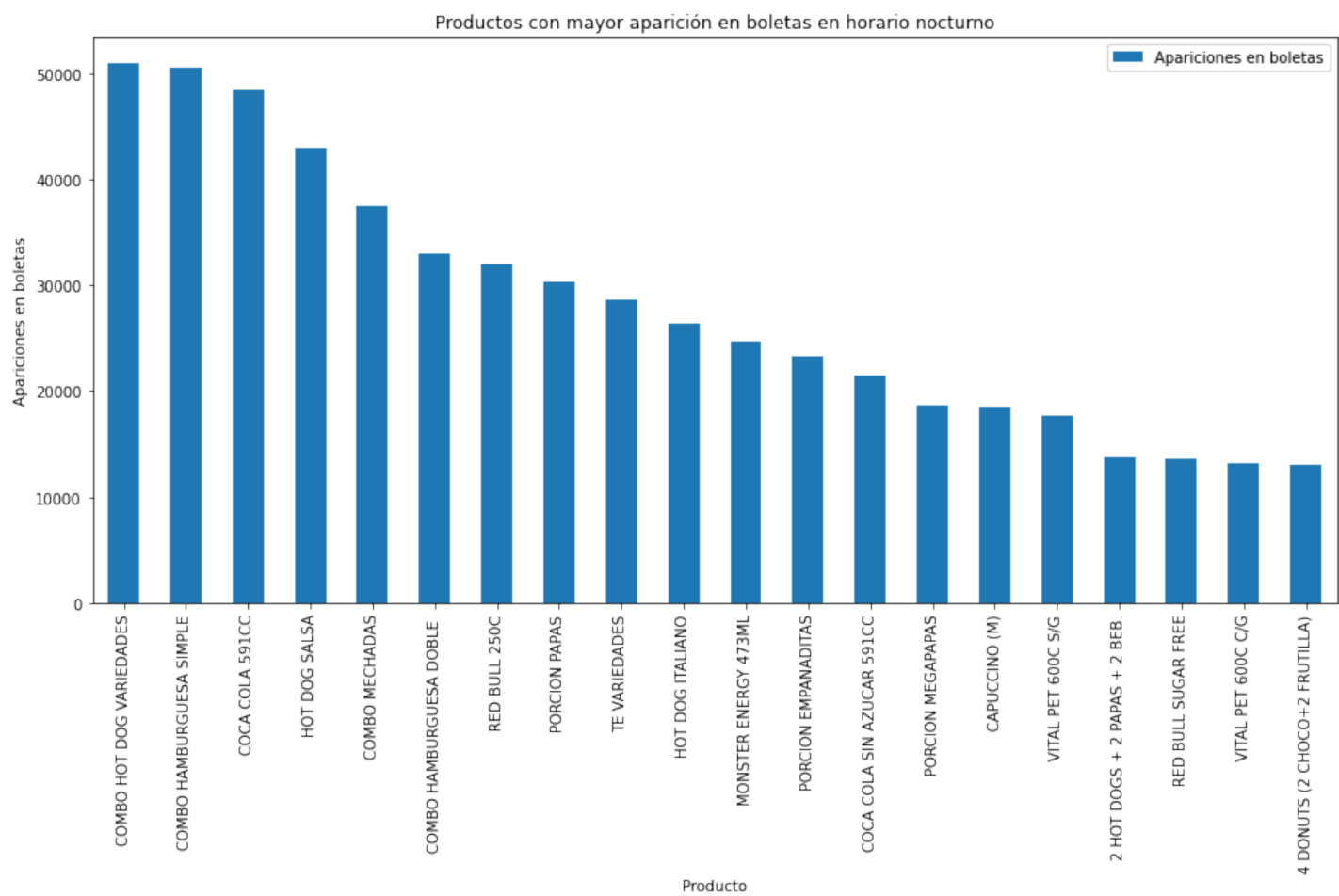


Figura 15: Productos con mayor aparición en boletas en horario de noche.
Fuente: Elaboración Propia.

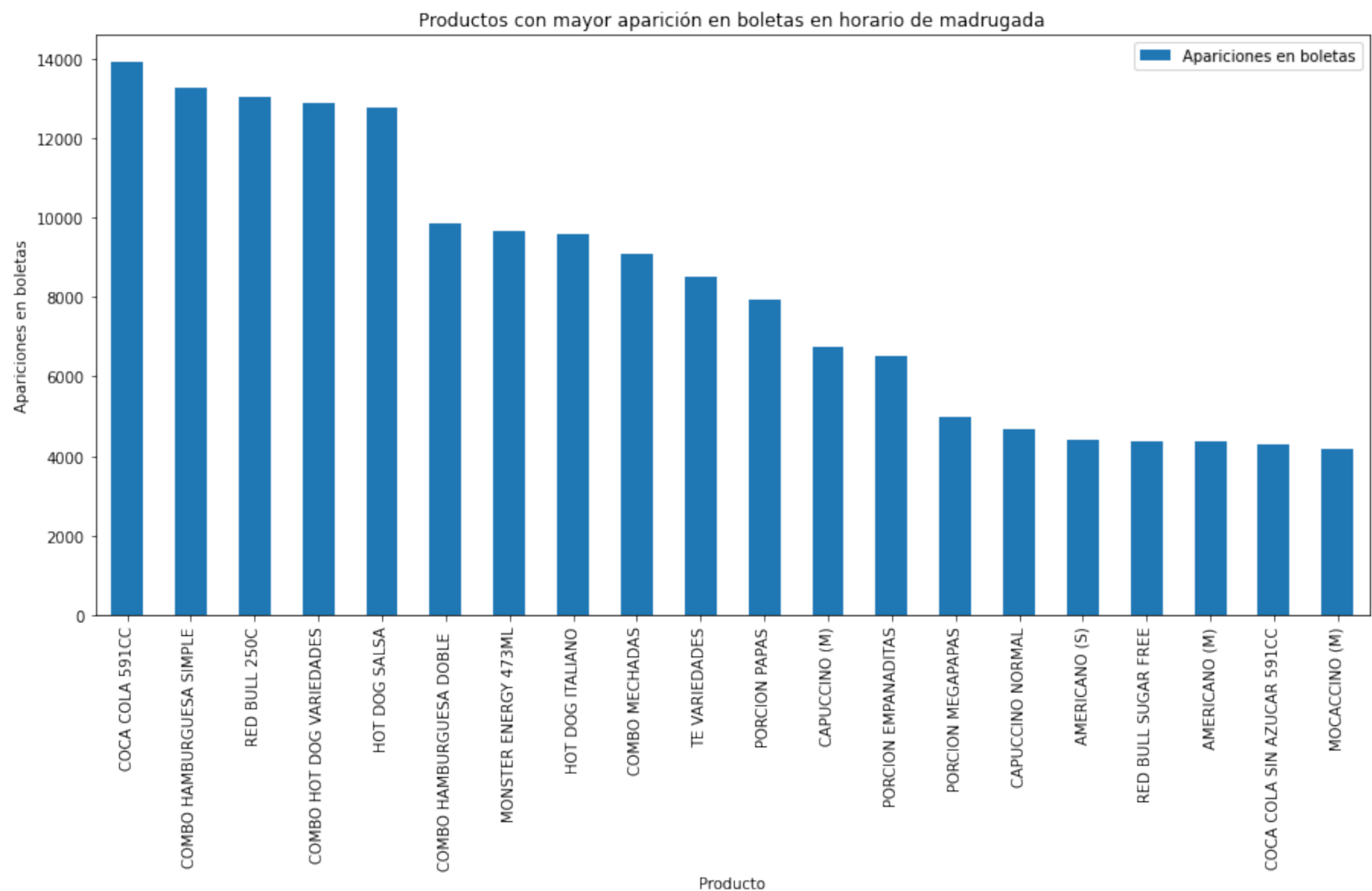


Figura 16: Productos con mayor aparición en boletas en horario de madrugada.
Fuente: Elaboración Propia.

En las figuras 17, 18, 19, 20 y 21 se puede ver los 20 productos con mayor aparición en boletas para las descripciones del tiempo de lluvias, cielos cubiertos, cielos nubosos, intervalos nubosos y cielos despejados respectivamente. Para el caso de la lluvia y cielos cubiertos, a diferencia de los productos con mayor aparición a nivel general (véase figura 8), aparece el combo de hamburguesa doble, la porción de papas y variedades de cafés. Para cielos nubosos aparecen solamente la porción de papas y un café americano; para cielos despejados aparece solamente una Cachantun, y para el caso de intervalos nubosos no hay productos que lo diferencien a nivel general.

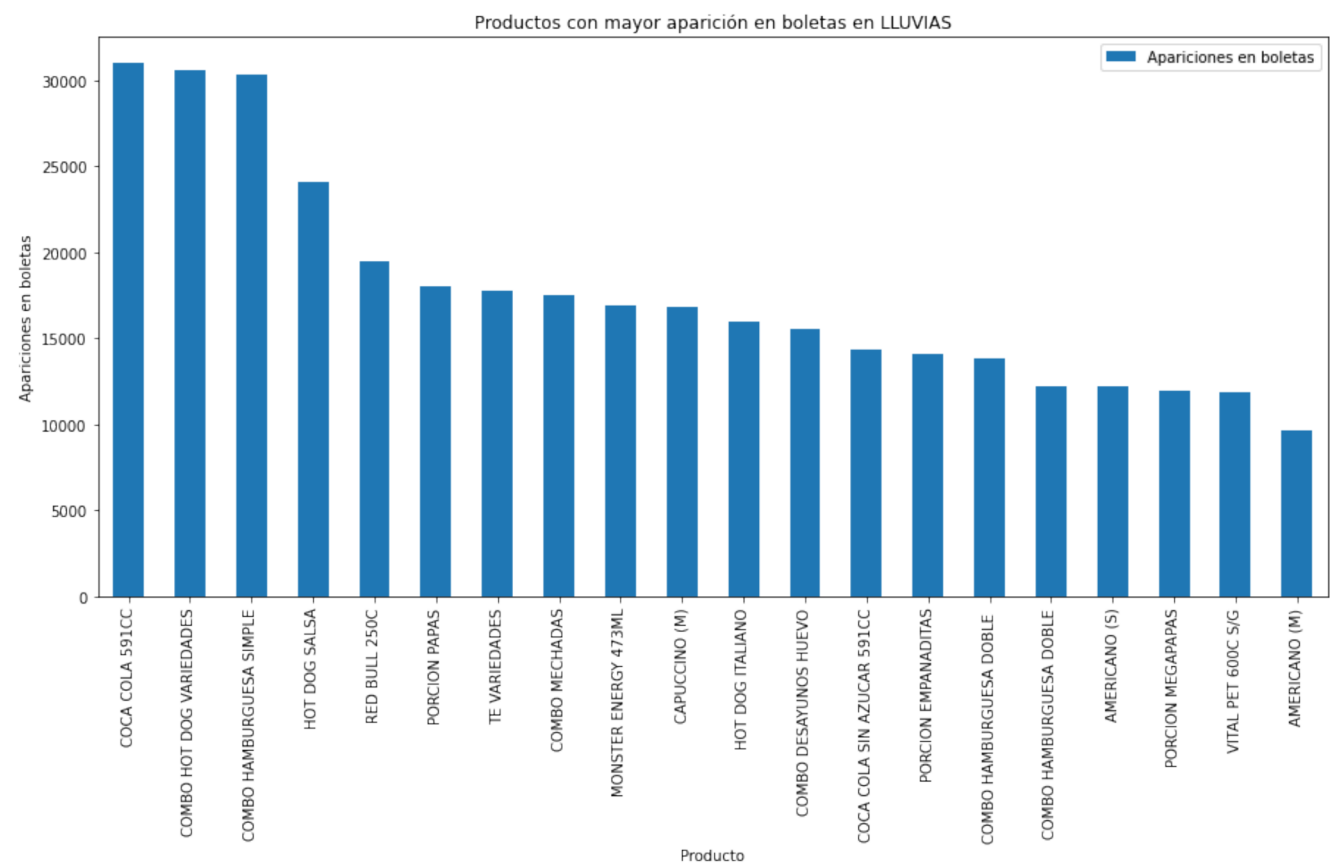


Figura 17: Productos con mayor aparición en boletas en lluvias.
Fuente: Elaboración Propia.

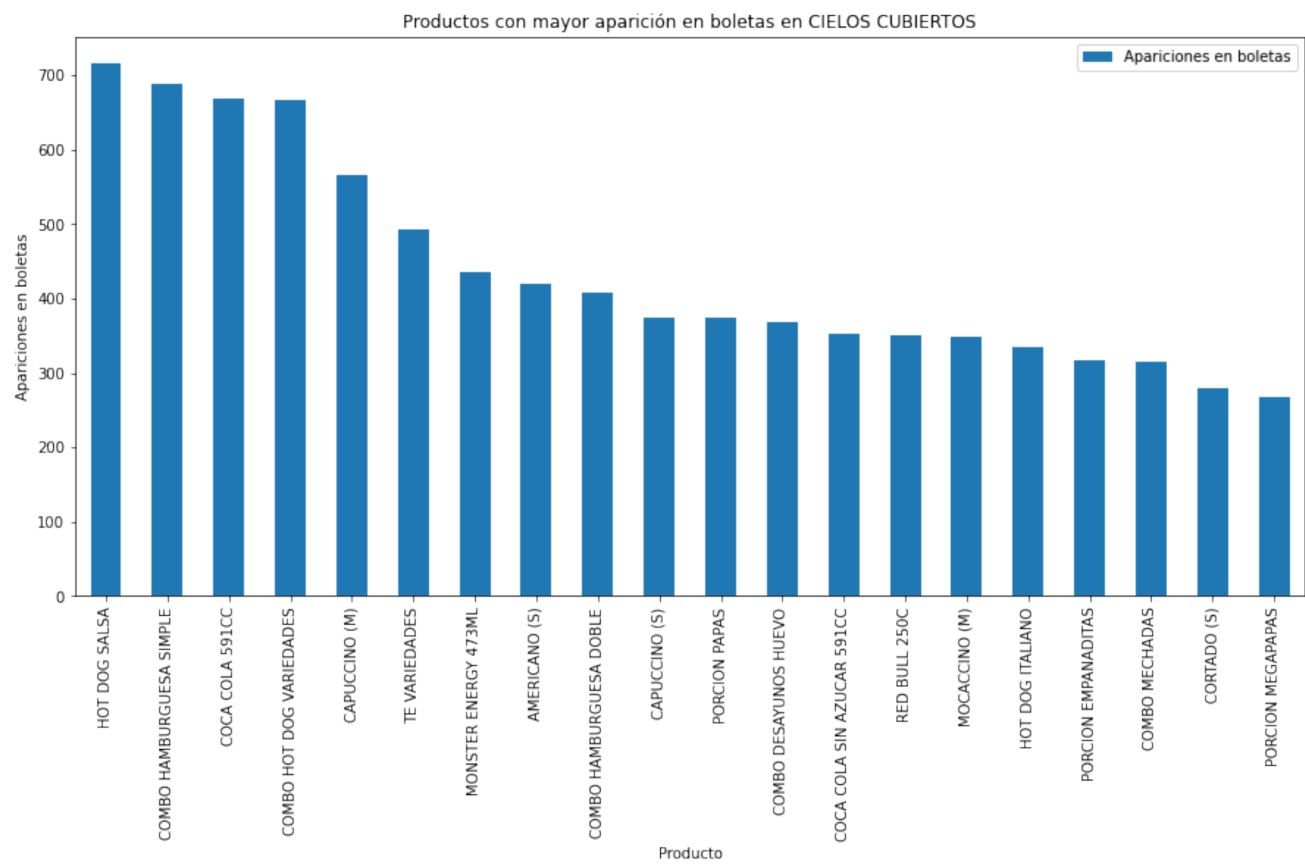


Figura 18: Productos con mayor aparición en boletas en cielos cubiertos.
Fuente: Elaboración Propia.

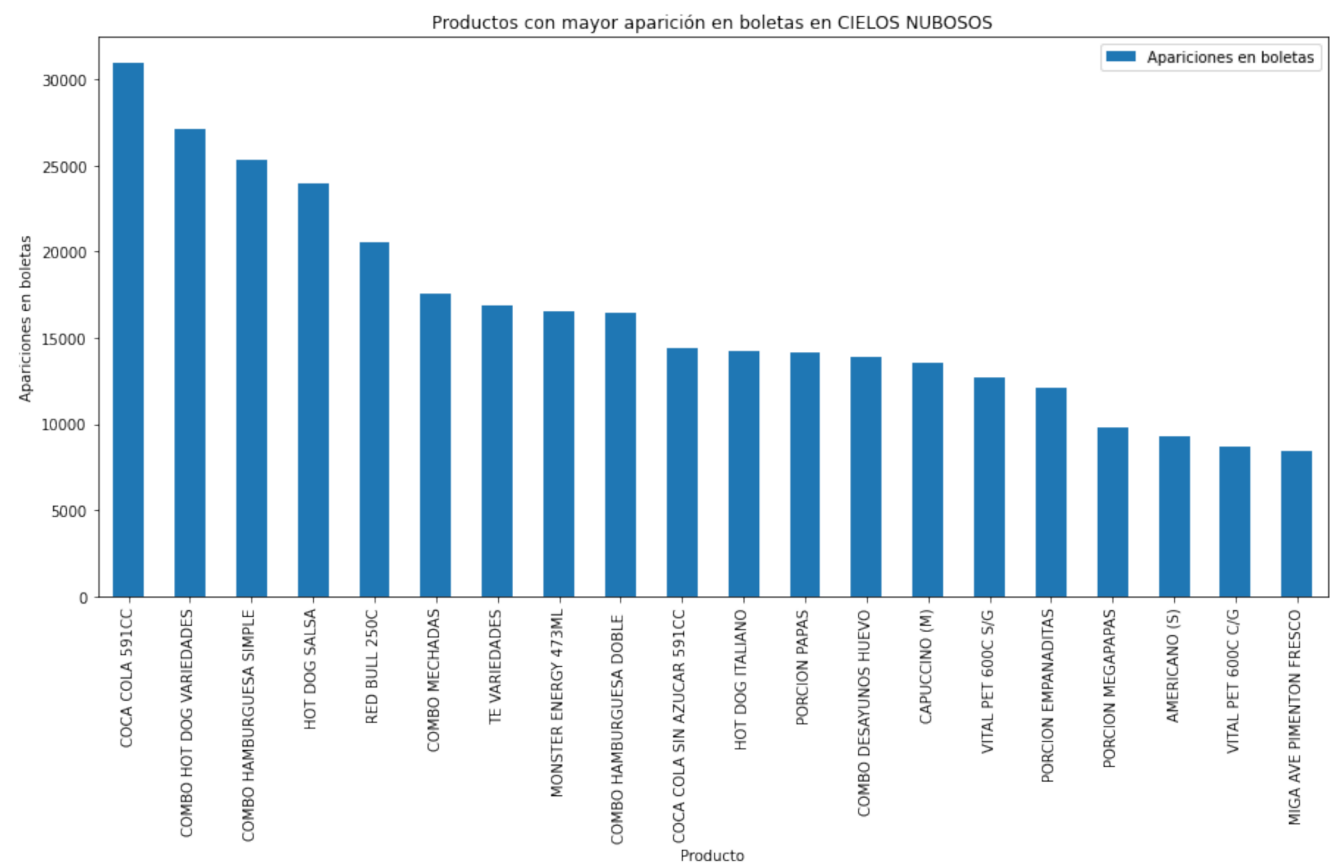


Figura 19: Productos con mayor aparición en boletas en cielos nublados.
Fuente: Elaboración Propia.

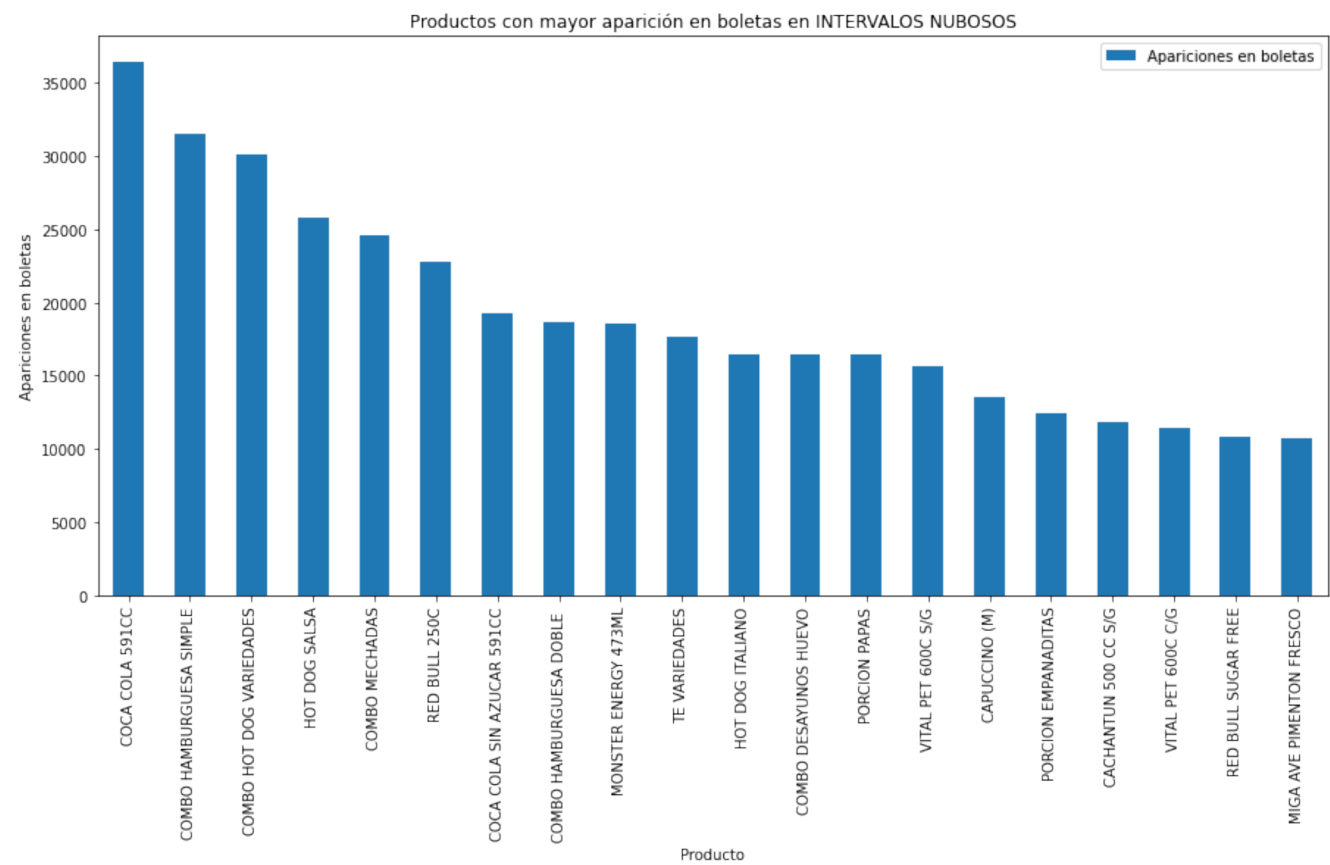


Figura 20: Productos con mayor aparición en boletas en intervalos nublados.
Fuente: Elaboración Propia.

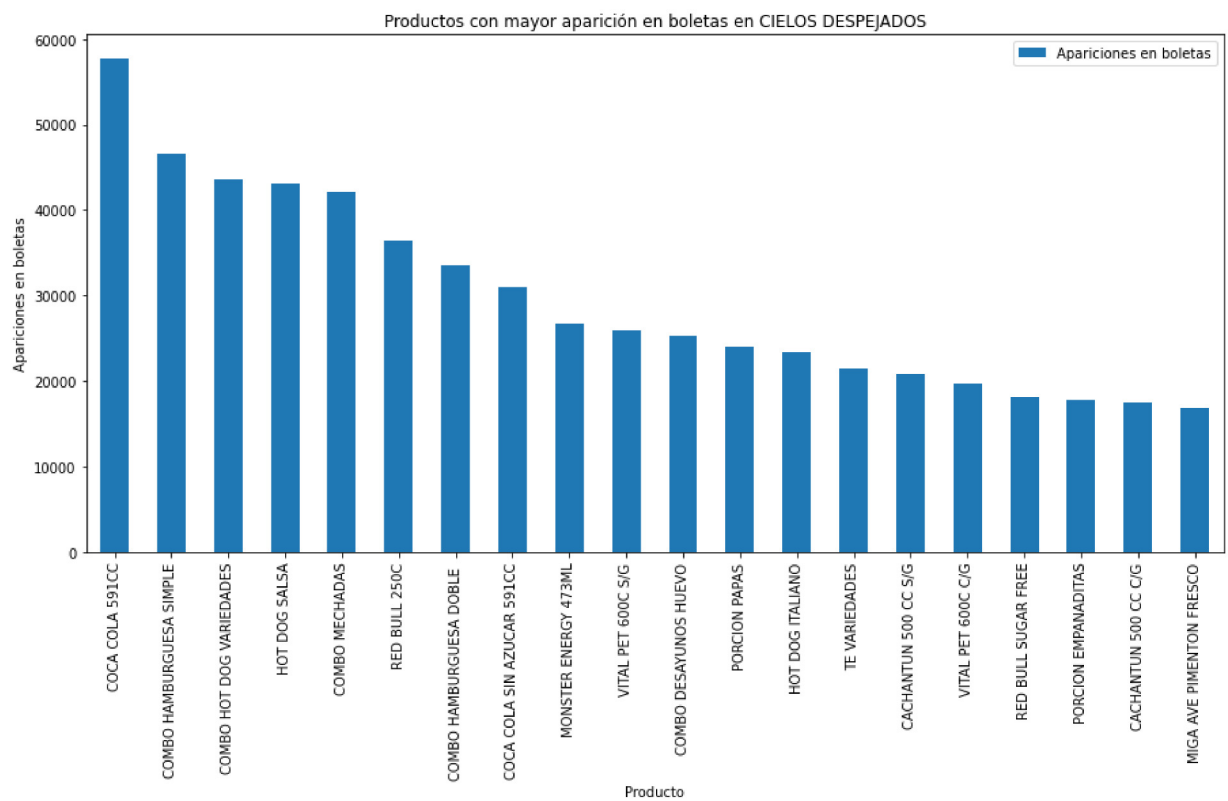


Figura 21: Productos con mayor aparición en boletas en cielos despejados.
Fuente: Elaboración Propia.

En las figuras 22 y 23 se pueden observar los 20 productos con mayor aparición en boletas para tiendas en formato carretera y ciudad, respectivamente. En el caso de las tiendas de carretera, los productos con mayor aparición son exactamente los mismo que a nivel general (véase la figura 8). Por otra parte, en el caso de las tiendas de ciudad, se puede ver que a diferencia de los productos con mayor aparición a nivel general, aparecen combos de desayuno tapadito, Coca Cola Light, cigarrillos y variedades de cafés.

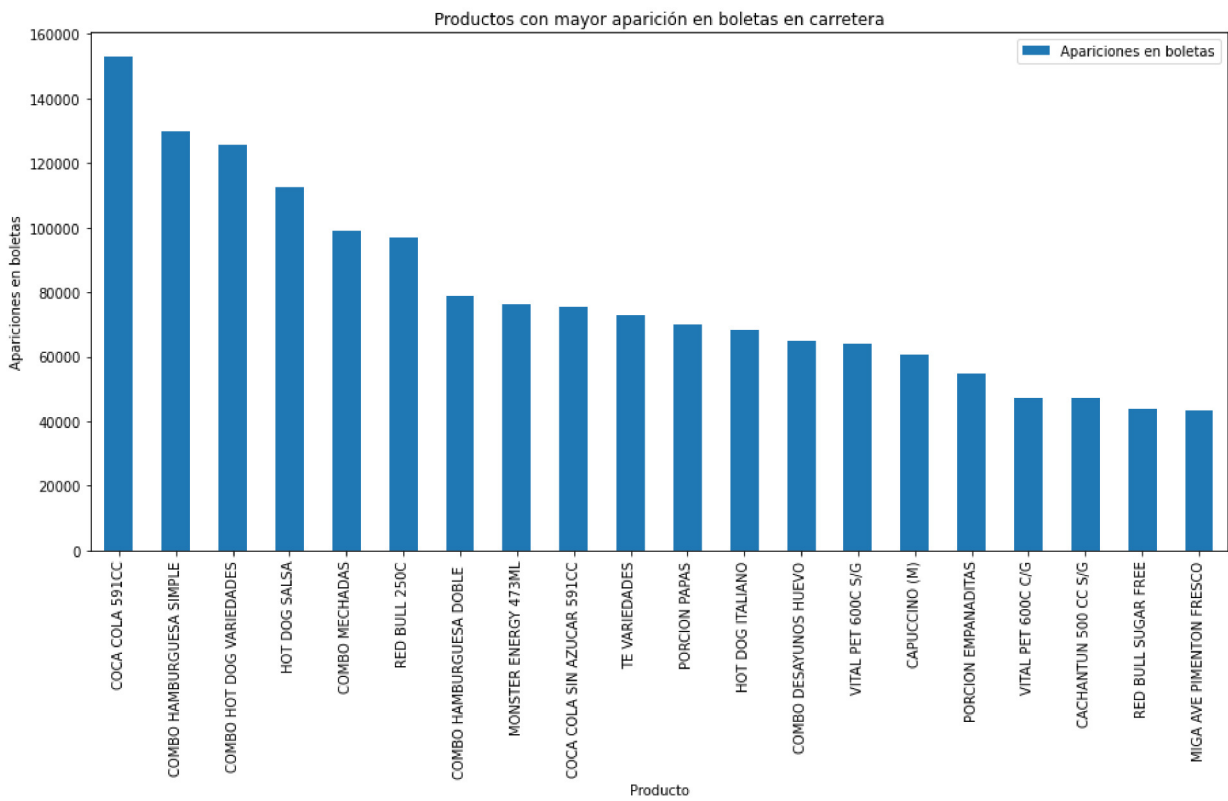


Figura 22: Productos con mayor aparición en boletas en carretera.
Fuente: Elaboración Propia.

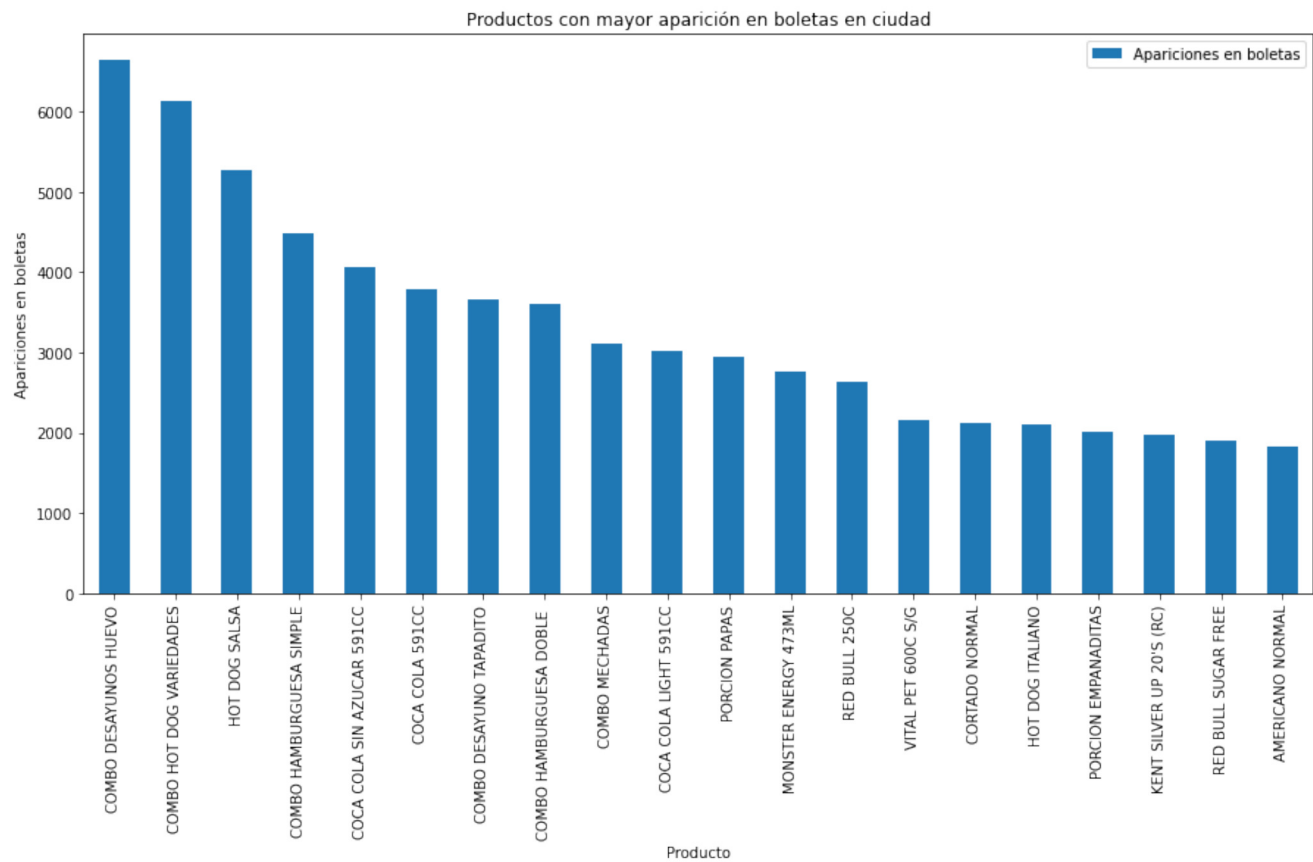


Figura 23: Productos con mayor aparición en boletas en ciudad.
Fuente: Elaboración Propia.

Observando los productos con mayor aparición en boletas sobre los distintos tipos de contextos, se puede concluir que el contexto no es un factor determinante para marcar tendencias de productos, ya que no existen grandes diferencias entre los productos que son populares a nivel global con los que los son a nivel de cada segmento.

Posterior a eso se realizó un análisis de canasta, con el objetivo de descubrir si existen pares de productos que poseen una tendencia a ser comprados juntos. Para ello se estudiaron las boletas y los productos que estas contienen. Se construyó una matriz en la cual las filas corresponden a las boletas, las columnas a los productos y los valores dentro de la matriz pueden ser 0 o 1, donde el valor será 0 si el producto no está presente en la boleta y 1 si es que sí aparece en ella.

Debido al alto volumen de transacciones y a la variedad de productos que existen, se estableció como regla que las combinaciones de productos cuenten con un Soporte Conjunto mayor o igual a 0.001. De las combinaciones que cumplen con esta regla, se seleccionaron las 3 con mayor Confianza.

Para reducir los grupos de transacciones y poder analizar las canastas por contexto, se segmentaron las transacciones por contextos cruzados, en los cuales se separaron las boletas según el momento del día en el que se realizaron y en el tipo de tienda en la que estas se emitieron. Para este estudio no se consideró el tiempo como factor para segmentar las boletas, ya que como se puede apreciar en las figuras 17 a 21, no existen muchas diferencias entre las tendencias de los distintos tiempos.

Tabla 4: Análisis de Canasta de Carretera-Mañana.

Fuente: Elaboración Propia.

Antecedente	Consecuente	Soporte	Confianza	Lift
DONUT FRUTILLA	DONUT CHOCOLATE	0.001547	0.262450	38.526571
MEDIALUNA UNITARIA	CORTADO (S)	0.002062	0.205966	15.934716
COMBO DESAYUNO TAPADITO	COMBO DESAYUNOS HUEVO	0.006991	0.199238	2.373204

Tabla 5: Análisis de Canasta de Carretera-Tarde.

Fuente: Elaboración Propia.

Antecedente	Consecuente	Soporte	Confianza	Lift
MENU PREMIUM C/ACOMPANAMIENTO	MENU NORMAL CON ACOMPANAMIENTO	0.004717	0.346725	15.161108
PORC NUGGETS NN 5UN	PORCION EMPANADITAS	0.001394	0.265850	14.046067
PORC NUGGETS NN 5UN	PORCION PAPAS	0.001226	0.233884	9.392866

Tabla 6: Análisis de Canasta de Carretera-Noche.

Fuente: Elaboración Propia.

Antecedente	Consecuente	Soporte	Confianza	Lift
PORC NUGGETS NN 5UN	PORCION EMPANADITAS	0.001982	0.294616	12.402133
DONUT FRUTILLA	DONUT CHOCOLATE	0.001395	0.276013	50.986821
MEDIALUNA UNITARIA	AMERICANO (S)	0.001227	0.210165	16.738657

Tabla 7: Análisis de Canasta de Carretera-Madrugada.

Fuente: Elaboración Propia.

Antecedente	Consecuente	Soporte	Confianza	Lift
NUGGETS	PORCION EMPANADITAS	0.001044	0.302564	13.751538
DONUT FRUTILLA	DONUT CHOCOLATE	0.001213	0.277284	52.433337
MEDIALUNA UNITARIA	AMERICANO (S)	0.001390	0.234347	15.135864

Tabla 8: Análisis de Canasta de Ciudad-Mañana.

Fuente: Elaboración Propia.

Antecedente	Consecuente	Soporte	Confianza	Lift
ESPRESSO CHICO	CAFE 9OZ/TE + 2 MLUNAS	0.001857	0.253731	10.998389
MEDIALUNA UNITARIA	CORTADO NORMAL	0.001835	0.205882	9.424059
LUCKY STRIKE BLUE/AZUL - HL20	ESPRESSO NORMAL	0.001398	0.181818	11.400747

Tabla 9: Análisis de Canasta de Ciudad-Tarde.

Fuente: Elaboración Propia.

Antecedente	Consecuente	Soporte	Confianza	Lift
PORCION EMPANADITAS	PORCION PAPAS	0.002492	0.172414	7.400541
MIGA AVE PIMENTON FRESCO	COCA COLA SIN AZUCAR 591CC	0.001015	0.109562	2.955603
VITAL PET 600C C/G	VITAL PET 600C S/G	0.001089	0.088060	5.112654

Tabla 10: Análisis de Canasta de Ciudad-Noche.

Fuente: Elaboración Propia.

Antecedente	Consecuente	Soporte	Confianza	Lift
DONUT FRUTILLA	DONUT CHOCOLATE	0.001002	0.224599	50.085561
M&M PLAIN 48GR.	M&M PEANUT 49GR.	0.001026	0.200000	25.563415
PORCION EMPANADITAS	PORCION PAPAS	0.003053	0.172973	6.394814

Tabla 11: Análisis de Canasta de Ciudad-Madrugada.

Fuente: Elaboración Propia.

Antecedente	Consecuente	Soporte	Confianza	Lift
M&M PLAIN 48GR.	M&M PEANUT 49GR.	0.001161	0.434783	91.304348
BARRA PROTEIN CHOCOLATE MANI	BARRA WILD PROTEIN CHOCOLATE	0.001394	0.387097	104.153226
KEM XTREME PET 1,5 L	MONSTER ENERGY 473ML	0.001045	0.333333	15.430108

Se puede observar que ninguna combinación cuenta con una confianza mayor a 0.5, por lo que se puede concluir que es más probable que los productos se lleven por separado a que se lleven juntos. Además, el soporte de estas combinaciones es bastante bajo, las cuales cuentan con una participación cercana al 0.1 % del total de boletas; sin embargo, esto puede deberse a la gran variedad de productos que existen y al alto volumen de boletas, ya que mientras mayor sea la cantidad de boletas, la probabilidad de que se repita una misma combinación de productos es menor.

Por otra parte, todas las combinaciones cuentan con un *lift* mayor a 1, por lo que sí existe una relación de dependencia entre estos productos. A pesar de que el Soporte Conjunto y la Confianza sean bajos, no se puede descartar el hecho de que existen usuarios puntuales que si compran estas combinaciones, lo que influirá a la hora de generar el modelo, ya que el sistema recomendador aprenderá a reconocer los intereses de los usuarios con consumo similar.

Dado que no existen grandes diferencias entre las tendencias de los distintos contextos y la baja confianza de las combinaciones de productos con mayor aparición en boletas, se llegó a la conclusión que, en base a los datos con los que se está trabajando actualmente, el contexto no es un factor determinante para que un usuario decida que productos va a comprar, por lo que se tomó la decisión de no implementar un sistema recomendador que sea sensible al contexto. Sin embargo, esto puede deberse a que sólo se están analizando 28 tiendas y que los productos no están bien estandarizados, ya que existen productos que se encuentran repetidos con distintos nombres, por lo que estos datos pueden estar ensuciando los resultados del análisis. Es por eso que en caso de trabajar con datos más limpios, es posible que al repetir el estudio, los resultados lleguen a ser distintos.

3.3.3. Preparación de los datos

El siguiente paso consistió en generar el set de datos con el cual se entrena al modelo, por lo que es necesario aplicar una serie de transformaciones a los datos en bruto, donde se aplican *queries* que ayuden a llevar las transacciones al formato requerido para el modelamiento. En la tabla 12 se muestra el formato de las transacciones preprocesadas

Tabla 12: Datos preprocesados.

Fuente: Elaboración Propia.

Atributo	Descripción
codigo_empresa_codigo_tx	Identificador único de boleta
id_usuario	Identificador único de usuario
codigo_producto	Identificador único de producto

En la tabla 13 se muestra el formato al cual se espera llevar los datos:

Tabla 13: Datos procesados.

Fuente: Elaboración Propia.

Atributo	Descripción
id_usuario	Identificador único de usuario
codigo_producto	Identificador único de producto
n_trx_producto	Cantidad de veces que el producto aparece en las boletas del usuario

Extracción de datos

En este paso se realizó una extracción de datos distinta a la que se hizo para el análisis de datos, ya que previamente se buscaba obtener solamente las transacciones que cumplan con las condiciones estipuladas. En este caso, se realizaron distintas consultas con el objetivo de armar dos *sets* de datos, los cuales se usarán para entrenar y probar el modelo. Para esto, se separaron las compras de los usuarios que realizaron más de 10 transacciones de los que compraron menos que esa cantidad. Se hace esta división para poder evaluar el modelo sobre usuarios que tengan múltiples transacciones, ya que mientras más información se tenga de un usuario, es más fácil identificar sus intereses. Posteriormente, se deja en el set de prueba el último producto que ha comprado uno de los usuarios con más de 10 transacciones, y el resto de las relaciones usuario-producto van al set de entrenamiento.

En el *listing 1* se identifican aquellos usuarios que han hecho más de 10 transacciones y los productos que llevaron en su última compra. Estos datos servirán para comprobar si el modelo es capaz de recomendar los productos que aparecen en las últimas compras.

En el *listing* 2 se identifican los usuarios que han hecho más de 10 transacciones, junto a los productos que han comprado, la cantidad de veces que los han llevado y la cantidad de transacciones que han realizado. Estos datos pasarán al set de entrenamiento. Esta consulta no contabiliza aquellos productos que aparecen en la última compra. Si existe un producto que fue comprado antes de la última compra pero que también se llevó en ella, esta no es considerada al sacar el total de veces que el usuario ha comprado el producto.

En el *listing* 3 se identifican los usuarios que han realizado 10 transacciones o menos, junto a los productos que han comprado, la cantidad de veces que los han llevado y la cantidad de transacciones que han realizado. Estos datos pasarán al set de entrenamiento.

En el *listing* 4 se identifican los usuarios que han hecho más de 10 transacciones, junto a todos los productos que han comprado, la cantidad de veces que los han comprado y la cantidad de transacciones que han realizado. A diferencia del *listing* 2, este si contabiliza las últimas transacciones e incluye productos que han sido comprados únicamente en la última transacción.

Filtrado de datos

Al realizar el análisis exploratorio de datos, se utilizaron los nombres y grupos de productos que aparecen en la tabla **trx_detalle_rl**. En este caso, se utilizan los datos del maestro de productos proveído por la *stakeholder*. De esta forma, se filtran las transacciones por aquellas en las cuales los SKUs de sus productos estén presentes en ese maestro de productos. Este filtro se realiza una vez que se han armado los sets de entrenamiento y de prueba, ya que el equipo en el que se está trabajando no cuenta con la capacidad para filtrar la consulta por una lista tan larga de objetos.

De igual forma, se utilizó el maestro de tiendas proveído por la *stakeholder* para poder filtrar las transacciones por aquellas que fueron realizadas en tiendas Pronto que estén trabajando con sistema Sckuba, ya que los SKUs del nuevo maestro coinciden con los que se trabajan en tiendas Sckuba. En los *listings* 1, 2, 3 y 4, se aplicó este filtro pasando una tupla con los códigos de sitios, los cuales corresponden a los identificadores únicos de cada tienda.

Por otra parte, debido al alto volumen de transacciones y al limitado espacio en memoria del equipo utilizado para el desarrollo del proyecto, se filtraron las transacciones para trabajar sólo con aquellas que fueron realizadas desde el 1 de Enero del 2022 hasta el 8 de Noviembre del mismo año. Este filtro se aplicó dentro de los *listings* 1, 2, 3 y 4.

Transformación de datos

En el capítulo 3 se mencionó que las transacciones tienen un formato que consta del identificador de la transacción, el identificador del usuario y el código del producto. A estas transacciones se le aplicaron transformaciones, de forma en que en vez de traer la transacción

en aquel formato, traiga el identificador del usuario, el código del producto y la cantidad de veces que el usuario ha comprado el producto. Esto se realiza contando cada vez que un producto ha aparecido en las boletas del usuario. Así, es posible saber cuántas veces el usuario ha interactuado con el producto y será posible identificar que tanto interés tiene en él.

3.3.4. Modelamiento

Para el desarrollo del sistema recomendador se debe seleccionar uno de los métodos vistos en el capítulo 2. Se descartó el sistema recomendador basado en contenido, ya que posee la desventaja de que no recomienda objetos con características distintas a los ya consumidos previamente. Por otra parte, se descartó el de filtrado colaborativo basado en memoria, debido a que no recomienda productos que usuarios similares no hayan consumido. Es por esto que se decide implementar un sistema recomendador de filtrado colaborativo basado en modelos, ya que nos permite predecir las valoraciones de todos los usuarios para todos los productos. Es por esto que se requiere llevar el set de datos inicial a un formato matricial para, posteriormente, descomponerla mediante una factorización matricial. El formato requerido por los modelos corresponde a una matriz de usuarios y productos, donde las filas corresponden a los usuarios, las columnas a los productos, y los valores de ella las interacciones entre los usuarios con los productos. El set de datos debe pasar un proceso de ingeniería de características, en el cual se definen cuales son las interacciones que pasarán a la matriz.

Por otra parte se debe seleccionar uno de los algoritmos estudiados en el capítulo 2 para poder modelar los datos, los cuales corresponden a SVD y ALS. Se elige el algoritmo que se ajuste mejor al tipo de feedback que se puede obtener a partir de los datos transaccionales.

CAPÍTULO 4

VALIDACIÓN DE LA SOLUCIÓN

En el presente capítulo se explica como se implementó de la propuesta de solución planteada y se justifica cómo se tomaron las decisiones que surgieron a lo largo del desarrollo del proyecto. Se comienza con las herramientas utilizadas para desarrollar el proyecto, para posteriormente especificar como fue que se abordó cada uno de los pasos de la planificación presentada en el capítulo anterior.

4.1. Desarrollo del Proyecto

En la presente sección se explica el desarrollo de cada uno de los pasos de la planificación planteada en el capítulo anterior.

4.1.1. Modelamiento

En la presente sección se explica el proceso de selección de atributos, el ajuste de los parámetros del modelo y cuál fue el criterio utilizado para seleccionar el algoritmo para el desarrollo del sistema recomendador.

Ingeniería de características

Como se mencionó en el capítulo 2, para desarrollar un sistema recomendador de filtrado colaborativo basado en modelos, es necesario contar con una matriz de usuarios y productos. Para generar el modelo que entregue los resultados más apropiados, se generan distintas opciones de matrices, donde los valores en ellas se basan en la cantidad de veces que el usuario ha comprado cada producto. Para generar las matrices de usuarios y productos, se modeló la relación entre los usuarios con sus respectivos productos de la siguiente forma:

1. **Reach:** El valor será el total de veces que un usuario ha comprado el producto dada la cantidad de transacciones que este ha realizado. El valor se moverá en un intervalo entre 0 y 1.
2. **Binaria:** El valor será 0 si el usuario nunca ha comprado el producto, y 1 si lo ha comprado al menos una vez.
3. **Interacciones:** El valor será la cantidad de veces que el producto ha aparecido en las transacciones del usuario, donde el valor será 0 si el usuario nunca ha comprado el producto.

Debido a que el set de entrenamiento ya cuenta con el número de interacciones, se aplican funciones para obtener el *reach* y el modelado binario. Para calcular el *reach*, como cada fila cuenta con el número de veces que el producto ha aparecido en boletas del usuario y con el total de boletas por usuario, se dividen estos valores, los cuales corresponden al numerador y denominador, respectivamente. Por otra parte, para el caso del modelado binario, como el set de datos sólo contiene los productos que han sido comprados y no los que no lo han sido, sólo basta con rellenar con una columna de 1. En la figura 24 se puede observar el formato final en el cual queda el set de entrenamiento.

	id_usuario	codigo_producto	total_trx_usuario	interacciones	reach	binario
0	PERSONAS_5271259	132624	26	6	0.230769	1
1	PERSONAS_4794012	132624	24	2	0.083333	1
2	PERSONAS_4235256	132624	10	2	0.2	1
3	PERSONAS_4758052	132624	40	2	0.05	1
4	PERSONAS_4353533	132624	51	1	0.019608	1
...	...	...	...	...	...	...
7455999	PERSONAS_2866581	172081	4	1	0.25	1
7456000	PERSONAS_4728034	171717	2	1	0.5	1
7456001	PERSONAS_557157	146448	5	1	0.2	1
7456002	PERSONAS_585293	173814	3	1	0.333333	1
7456003	PERSONAS_868733	148255	2	1	0.5	1

Figura 24: Set final de entrenamiento.
Fuente: Elaboración Propia.

Como un usuario no suele consumir todos los productos disponibles, la mayoría de los valores en las matrices serán 0, lo cual se conoce como matriz *sparse*. Además, al trabajar con 976.202 usuarios y 2.653 productos, se generan matrices muy costosas de procesar, lo cual puede implicar en que se consuma mucha memoria del equipo y en que este se vea incapaz de procesarlas. Por otra parte, el set de entrenamiento sólo cuenta con los productos que han sido comprados, pero no contiene las relaciones de los productos que no han sido comprados por los usuarios. Es por ello que se utiliza el módulo *sparse* de la biblioteca *scipy*, donde se lleva el *dataframe* de *pandas* a una *csr_matrix*, la cual corresponde matriz *sparse* en formato comprimido, la cual incluye la información de los productos que no han sido comprados. Así, se generan 3 matrices *csr_matrix*, cada una con los valores de las columnas interacciones, *reach* y binario. Posterior a esto, a cada una de ellas se les aplica un algoritmo SVD o ALS para la generación del sistema recomendador.

Implementación de Alternating Least Square (ALS)

Como se mencionó en el capítulo 2, el algoritmo SVD es altamente utilizado para el desarrollo de sistemas recomendadores debido a su alta exactitud y escalabilidad. Sin embargo,

este algoritmo necesita *feedback* explícito, lo cual no se adecúa a los datos transaccionales o *feedback* implícito con el que se está trabajando actualmente. Es por esto que se optó por implementar el algoritmo ALS propuesto por [Hu *et al.*, 2008], el cual, a diferencia del algoritmo SVD, integra niveles de confianza en su función de costos. Los niveles de confianza representan qué tanta seguridad se tiene de que a un usuario le gusta un producto, ya que con los datos transaccionales, no se puede corroborar si a una persona le gusta un producto o no. Sin embargo, se puede saber cuántas veces ha comprado el producto y, a partir de esta información, inferir que tanto interés tiene el usuario sobre él.

Para facilitar la implementación de este algoritmo, se utilizó la biblioteca *implicit*, la cual provee implementaciones rápidas de distintos algoritmos para sistemas recomendadores basados en *feedback* implícito. Se utiliza la función *AlternatingLeastSquares* para generar el sistema recomendador, función basada en el algoritmo ALS propuesto por [Hu *et al.*, 2008].

Para comenzar con un modelo *baseline*, se constuyeron los modelos de las 3 matrices con los mismos parámetros, los cuales corresponden a α igual a 1, el número de factores latentes igual a 100 y un *random state* igual a 0. Debido a la extensión de este proyecto, no se alcanzó a probar los modelos con otros parámetros, por lo cual existen oportunidades de mejora para la optimización de hiperparámetros.

Los modelos retornan las nuevas matrices reconstruidas, las cuales contienen las estimaciones de cada producto para cada usuario. Los valores dentro de la matriz no significan nada por sí solos, sino que adquieren valor al compararlos entre sí mismos para un mismo usuario. Mientras mayor sea el valor estimado del producto para un usuario, significa que el producto es de mayor agrado para el usuario; por otra parte, mientras menor sea el valor, quiere decir que el producto será de menor agrado para el usuario. La escala de estos valores es propia para cada usuario. A partir de estas estimaciones, se evaluará el desempeño de los modelos.

4.1.2. Evaluación

Para poder determinar si las recomendaciones entregadas por el modelo son adecuadas, es necesario evaluar su desempeño. Para ello, se requiere la opinión de expertos que puedan corroborar si los resultados son apropiados y si se cumple con las necesidades del negocio. Se medirá el rendimiento del modelo de forma cuantitativa y cualitativa. Para medir el rendimiento de forma cuantitativa, se seleccionará una métrica de desempeño para sistemas recomendadores que se adapten al tipo de *feedback* extraído desde las transacciones.

Por otra parte, para medir el desempeño de forma cualitativa, se realizará un experimento en usuarios reales, los cuales contarán con distintos tipos de consumo en tiendas Pronto. Se espera poder mostrar las recomendaciones de productos personalizados para cada usuario y que puedan indicar si las recomendaciones son de su agrado o no.

Evaluación Cuantitativa: Métrica de desempeño

Para evaluar modelos basados en *feedback* implícito que sólo cuentan con valoraciones positivas, [Hu *et al.*, 2008] y [Johnson, 2014] proponen utilizar una métrica llamada *Mean Ranking Percentage*, la cual evalúa la satisfacción del usuario mediante una lista de objetos recomendados. A esta lista se le genera un ranking basado en percentiles, donde el objeto más recomendado tendrá un ranking igual a 0 y el objeto menos recomendado tendrá un ranking igual a 100. El ranking es distribuido uniformemente entre el resto de los objetos de la lista de recomendaciones. La métrica se define de la siguiente forma:

$$MRP = \frac{\sum_{ui} r_{ui}^t rank_{ui}}{\sum_{ui} r_{ui}^t}$$

donde:

- r_{ui} : Interés del usuario u por el producto i .
- $rank_{ui}$: Ranking del producto i para el usuario u .

donde el interés del usuario se verá representados por las tres variables en las cuales se basa el modelo, el *reach*, la variable binaria y la cantidad de interacciones.

Si el MRP es cercano a 0, significa que los productos que consume el usuario tienen un ranking bajo y tienen una mayor tendencia a ser recomendados. Por otra parte, si el MRP es más cercano a 1, significa que los productos consumidos tienen un ranking alto y por ende no tienden a ser recomendados.

Para evaluar los modelos, se utiliza una muestra que contiene las predicciones de todos los productos de aquellos usuarios que han hecho más de 10 transacciones, junto a la cantidad de veces que han comprado el respectivo producto. De esta forma, se evaluó el MRP sobre los 3 modelos generados, donde se obtuvieron los siguientes resultados:

- $MRP_{reach} = 0,067$
- $MRP_{binario} = 0,058$
- $MRP_{interacciones} = 0,039$

El modelo basado en la cantidad de interacciones que tienen los usuarios con los productos es el que cuenta con un menor MRP, por lo que se será seleccionado para pasar a la siguiente fase de experimentación sobre usuarios reales.

Evaluación Cualitativa: Pruebas en usuarios reales

Previamente se evaluaron cuantitativamente los tres modelos generados; sin embargo, la métrica utilizada mide qué tan bien se posicionan los productos comprados por aquellos usuarios que han realizado más de 10 transacciones, por lo que no se sabe como se comportará el sistema con aquellos usuarios que han hecho menos transacciones, ni tampoco se puede determinar si los productos que no han sido comprados pueden llegar a ser del agrado de los usuarios. Es por ello que se planificó un *workshop*, con el objetivo de probar qué tan adecuadas son las recomendaciones del sistema recomendador sobre usuarios reales. Para ello, se determinaron casos de uso para el sistema recomendador y se plantearon distintas hipótesis, en las cuales los usuarios tuvieron que responder si para ellos se cumplieron o no. A cada usuario se le entregó una planilla por cada caso de uso con los resultados personalizados entregados por el sistema recomendador, además de una planilla con los productos que ha consumido y la cantidad de veces que los ha comprado.

El primer caso de uso del sistema recomendador consiste en personalizar el orden en que se muestran las categorías en Escanea & Paga. El sistema recomendador genera estimaciones acerca de qué tanto le puede llegar a gustar cada producto a cada usuario, por lo que para generar un orden de categorías personalizadas, se calculan las estimaciones de cada producto, se agrupan por cada categoría y se obtiene la mediana de cada una de las categorías. Así, las categorías se ordenan de mayor a menor mediana. Se le mostró a cada usuario el orden de categorías preestablecido que existe en la aplicación y el orden personalizado. Se plantearon dos hipótesis, la primera consiste en que el orden personalizado tiene sentido en base a los productos que compra el usuario, y la segunda plantea que el orden personalizado es de mayor agrado que el orden preestablecido. En las figuras 25 y 26 se puede visualizar la distribución de las respuestas de los usuarios: El 80 % de los usuarios opinaron que el orden personalizado tiene sentido en base al consumo que ellos poseen, y el 100 % de los usuarios prefirieron el orden personalizado sobre el orden preestablecido.

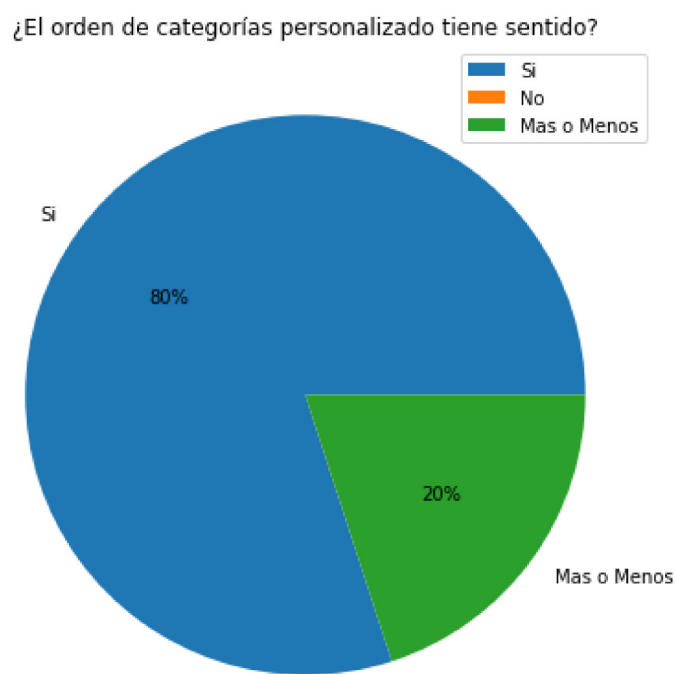


Figura 25: Encuesta sobre el sentido del orden de categorías personalizado.
Fuente: Elaboración Propia.

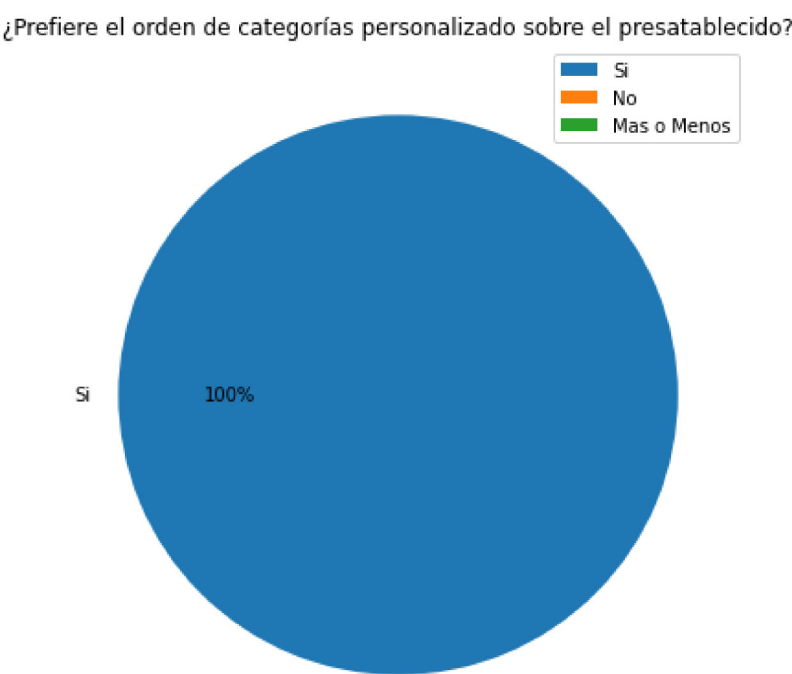


Figura 26: Encuesta sobre preferencia sobre el orden de categorías personalizado.
Fuente: Elaboración Propia.

El segundo caso de uso del sistema recomendador consiste en personalizar el orden en que aparecen los productos dentro de las categorías del catálogo de Escanea & Paga. Similar al caso anterior, se obtienen las estimaciones de los productos para cada usuario y se ordenan de mayor a menor estimación dentro de cada categoría. Para cada usuario, se seleccionaron las tres primeras categorías obtenidas en el caso de uso anterior y se le mostraron los productos ordenados de manera personalizada. Las hipótesis planteadas consisten en que el orden de los productos personalizados tiene sentido en base al consumo realizado por el usuario, y que el orden de productos personalizado es de mayor agrado que el orden preestablecido. En las figuras 27 y 28 se puede visualizar la distribución de las respuestas de los usuarios: El 60 % de los usuarios opinaron que el orden de productos personalizado tiene sentido en base a su propio consumo y el 50 % de los usuarios prefirieron el orden personalizado sobre el orden preestablecido.

¿El orden de productos personalizado tiene sentido?

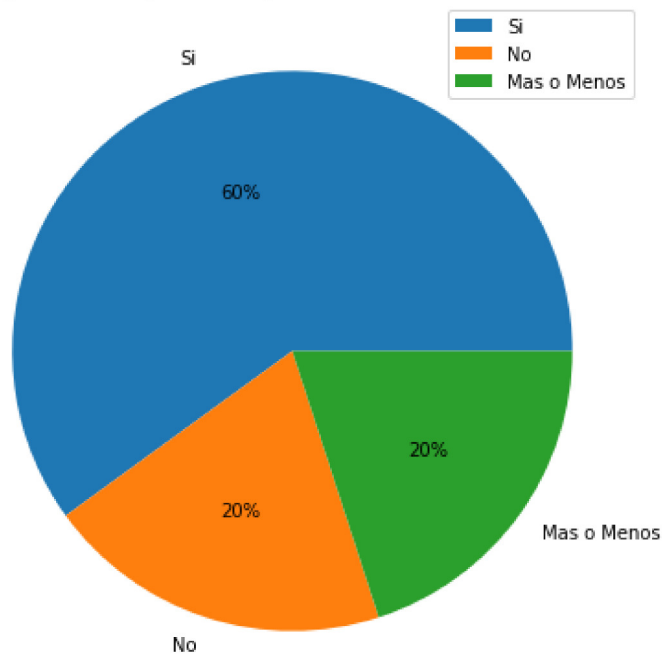


Figura 27: Encuesta sobre el sentido del orden de productos personalizado.
Fuente: Elaboración Propia.

¿Prefiere el orden de productos personalizado sobre el presatablecido?

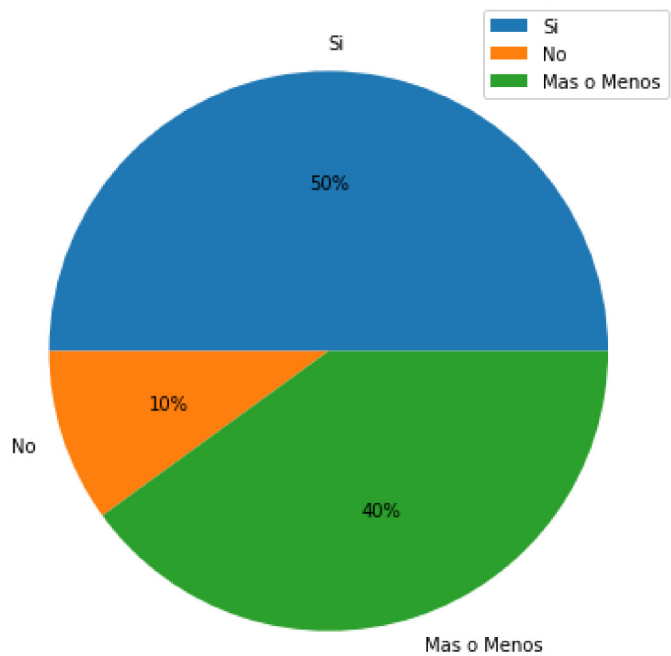


Figura 28: Encuesta sobre la preferencia sobre el orden de productos personalizado.
Fuente: Elaboración Propia.

El tercer y cuarto caso de uso del sistema recomendador consiste en obtener recomendaciones de productos sustitutos y productos complementarios. Es posible detectar productos similares entre sí en base a los factores latentes obtenidos a través del sistema recomendador, mientras menor sea la distancia entre los factores latentes, mayor será la similitud. Los productos son considerados como similares si los usuarios que los consumen compran productos de manera similar. Los productos se denominan como sustitutos si son similares y pertenecen a una misma categoría; por otra parte, dos productos se denominan como complementarios si son similares pero pertenecen a distintas categorías. Para probar estos casos de uso, se seleccionó el producto más comprado por cada usuario, y en base a este se le recomendaron sus productos sustitutos y complementarios. Para ambos casos de uso se usaron las mismas hipótesis, que cada una de las recomendaciones tiene sentido en base al producto con el que se está comparando y que cada recomendación es del agrado de su respectivo usuario. En las figuras 29 y 30 se puede observar la distribución de las respuestas obtenidas para los experimentos de recomendaciones de productos sustitutos y complementarios respectivamente. Para el caso de las recomendaciones de productos sustitutos, el 93 % de las recomendaciones tienen sentido en base a los productos con los que se hizo la comparación, y el 77 % de las recomendaciones fueron del agrado de los usuarios. Por otra parte, para el caso de los productos complementarios, el 73 % de las recomendaciones tienen sentido en base a los productos con los que se hizo la comparación y el 60 % de las recomendaciones fueron del agrado de los usuarios.

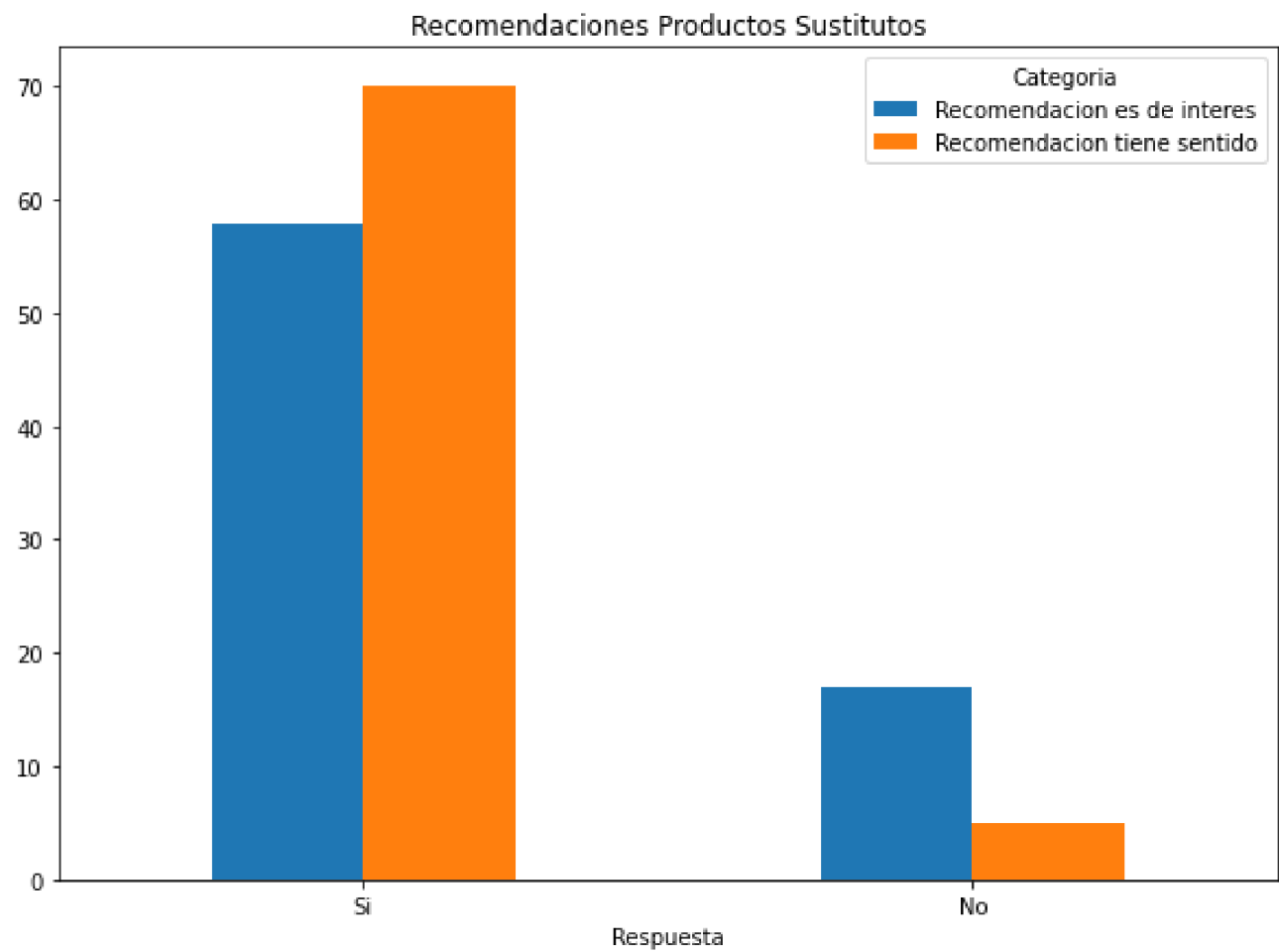


Figura 29: Encuesta sobre Recomendaciones de Productos Sustitutos.
Fuente: Elaboración Propia.

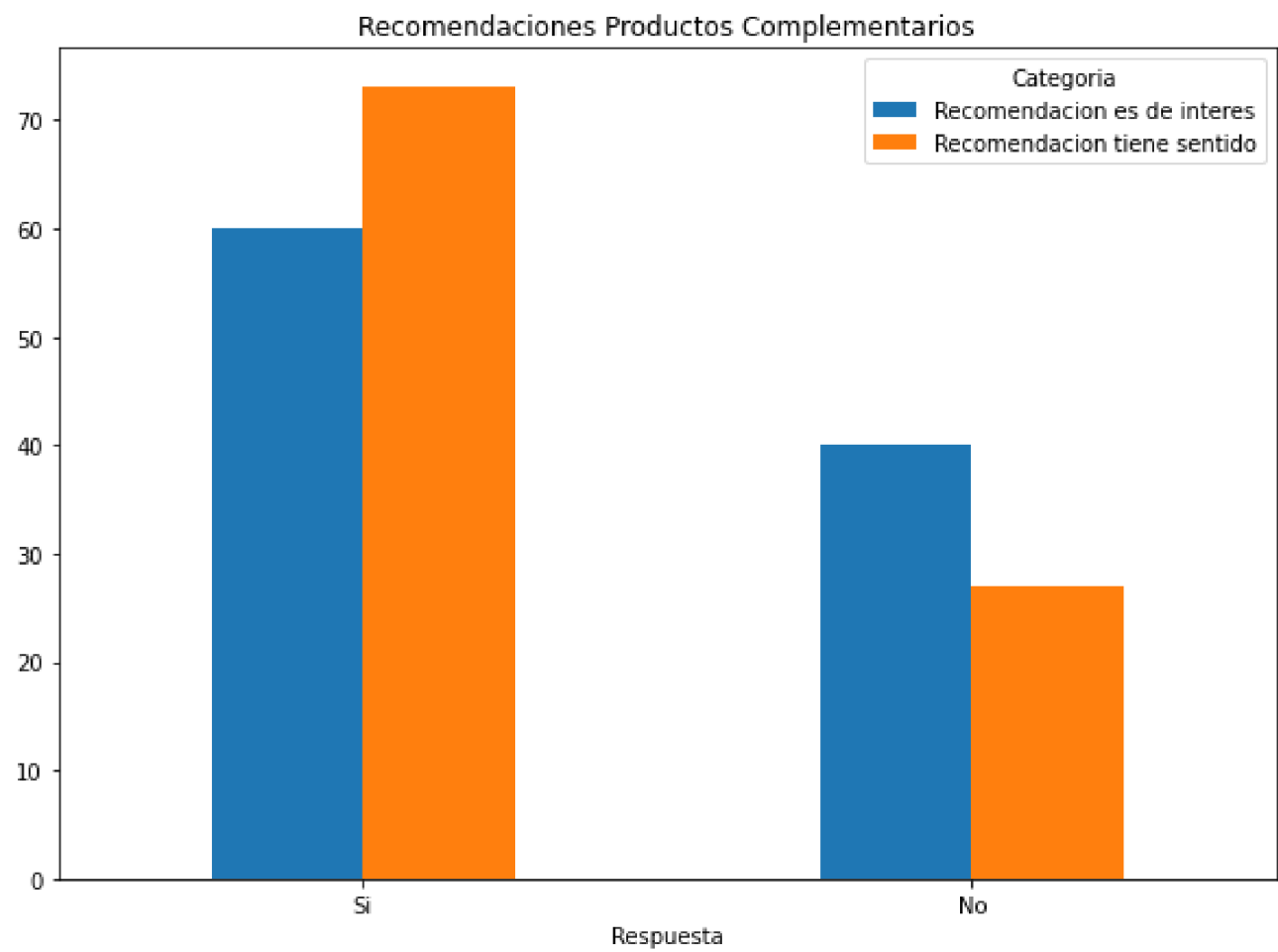


Figura 30: Encuesta sobre Recomendaciones de Productos Complementarios.
Fuente: Elaboración Propia.

El quinto caso de uso del sistema recomendador consiste en entregar recomendaciones personalizadas para los usuarios. El sistema recomendador mostrará un listado de 10 productos para cada usuario, donde los productos seleccionados serán aquellos con mayor estimación para cada uno de ellos, ordenados de mayor a menor preferencia. Las hipótesis planteadas consisten en que cada una de las recomendaciones tiene sentido en base a los productos que consume el usuario y que cada recomendación es del agrado del usuario. A cada uno de los usuarios de prueba se les entregó una planilla con sus 10 productos recomendados, donde por cada producto recomendado se respondió si se cumplen o no las hipótesis planteadas. En la figura 31 se puede observar la distribución de las respuestas obtenidas en este experimento. El 90 % de las recomendaciones tuvieron sentido en base al consumo de los usuarios y el 78 % de las recomendaciones fueron de agrado para los usuarios.

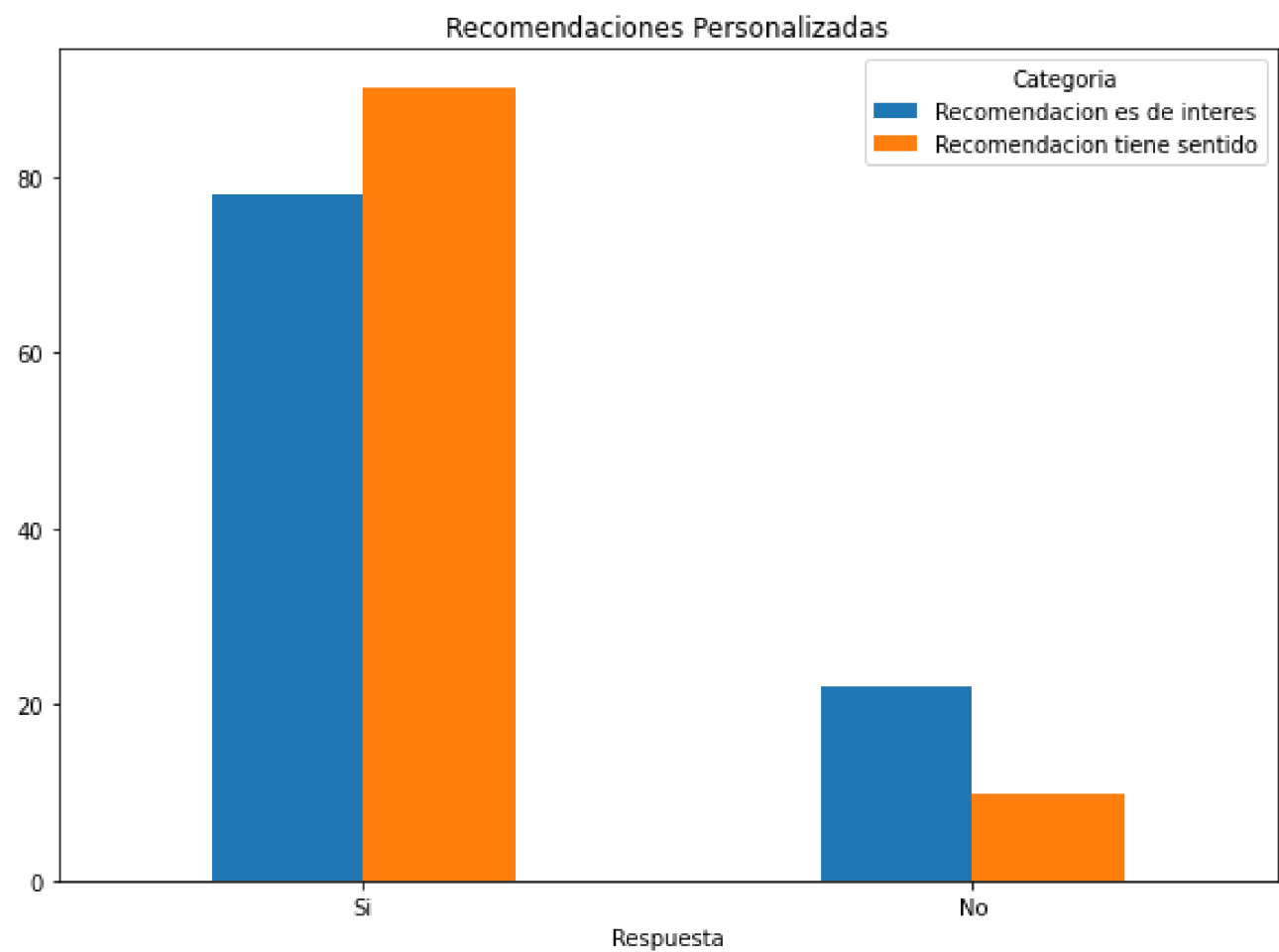


Figura 31: Encuesta sobre Recomendaciones Personalizadas.
Fuente: Elaboración Propia.

El sexto y último caso de uso del sistema recomendador consiste en entregar recomendaciones de usuarios similares a uno mismo. Previamente se detectaron productos similares en base a la distancia entre sus factores latentes. Como usuarios y productos se ven caracterizados de igual forma por medio de factores latentes, también es posible identificar usuarios que son similares entre sí. Para cada usuario de prueba, se obtuvieron los 5 usuarios más similares a este mismo, se generaron las recomendaciones personalizadas para los 5 usuarios y se ordenaron los productos obtenidos en base al número de apariciones sobre estas recomendaciones. Las hipótesis planteadas consisten en que cada una de las recomendaciones tiene sentido en base a los productos que consume el usuario y que cada recomendación

es del agrado del usuario. A cada usuario de prueba se le entregó una planilla con las recomendaciones obtenidas a partir de sus respectivos usuarios similares, y para cada producto recomendado respondieron si las hipótesis se cumplieron o no. En la figura 32 se puede observar la distribución de las respuestas obtenidas en este experimento. En este caso, el 97 % de las recomendaciones tuvieron sentido en base al consumo de los usuarios y el 92 % de las recomendaciones fueron de agrado para los usuarios.

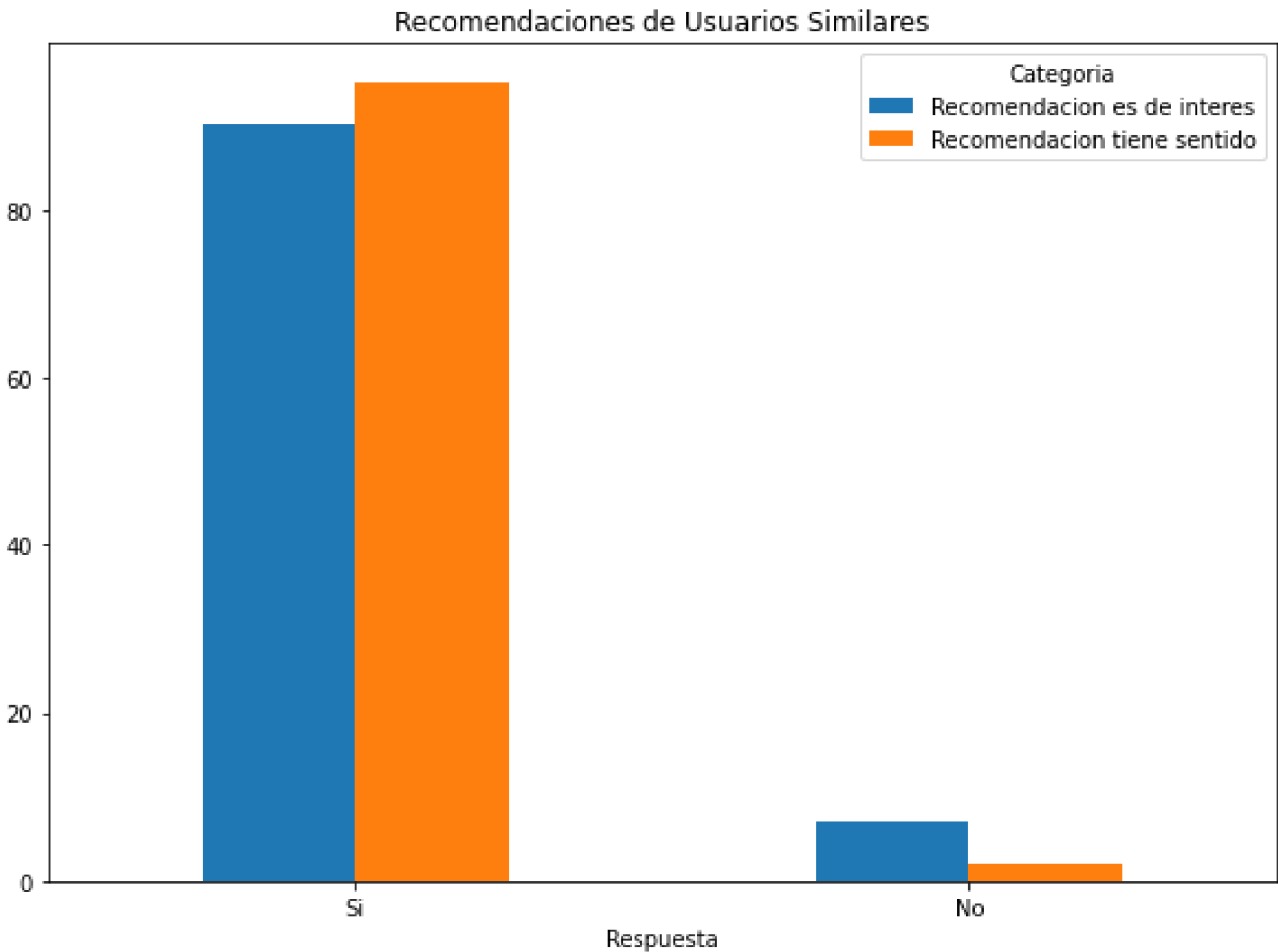


Figura 32: Encuesta sobre Recomendaciones de Usuarios Similares.
Fuente: Elaboración Propia.

4.1.3. Despliegue

En la presente sección, se explica el posible plan para implementar el sistema recomendador en la aplicación Muevo. Debido a la extensión del proyecto, no fue posible desplegar el sistema dentro de la aplicación; sin embargo, esto fue una prueba de concepto, por lo que, independiente de si se decide continuar o no con el, no es seguro que se lleve a la aplicación. Actualmente se busca comparar el rendimiento del sistema recomendador desarrollado con el de *Amazon Personalize*, el servicio de recomendaciones proveído por *Amazon Web Services*.

Casos de uso

En la sección anterior se probaron distintos casos de uso para el sistema recomendador y se

buscó conocer la satisfacción de los usuarios por medio de las recomendaciones entregadas; sin embargo, no se consideró el cómo interactuarían los usuarios con el sistema recomendador dentro la aplicación. Si bien esta es una labor para el equipo de Experiencia Usuaria de la gerencia de Productos Digitales, se decidió hacer una propuesta acerca de cómo llevar el sistema recomendador a la interfaz de Escanea & Paga.

Para el orden personalizado del catálogo, el cambio que se debería hacer en la interfaz consiste en reordenar el orden en el que aparecen las categorías en el catálogo de Escanea & Paga, donde en lugar de aparecer en un orden preestablecido, las categorías pasarían a tener el orden personalizado generado por el sistema recomendador. En la figura 33 se puede observar que este cambio modificaría la barra superior de búsqueda en la que aparecen las categorías, sin embargo también modificaría la forma en que se presentan las categorías verticalmente.



Figura 33: Cambios en la interfaz dado el orden personalizado de categorías.
Fuente: Elaboración Propia.

De manera similar, para el caso del orden personalizado de productos, el único cambio en la interfaz consiste en modificar el orden preestablecido de los productos, por el orden personalizado generado por el sistema recomendador. En la figura 34 se puede observar la sección de la interfaz a la cual se aplicaría el cambio, la cual corresponde a los productos que aparecen dentro de las categorías del catálogo de Escanea & Paga.

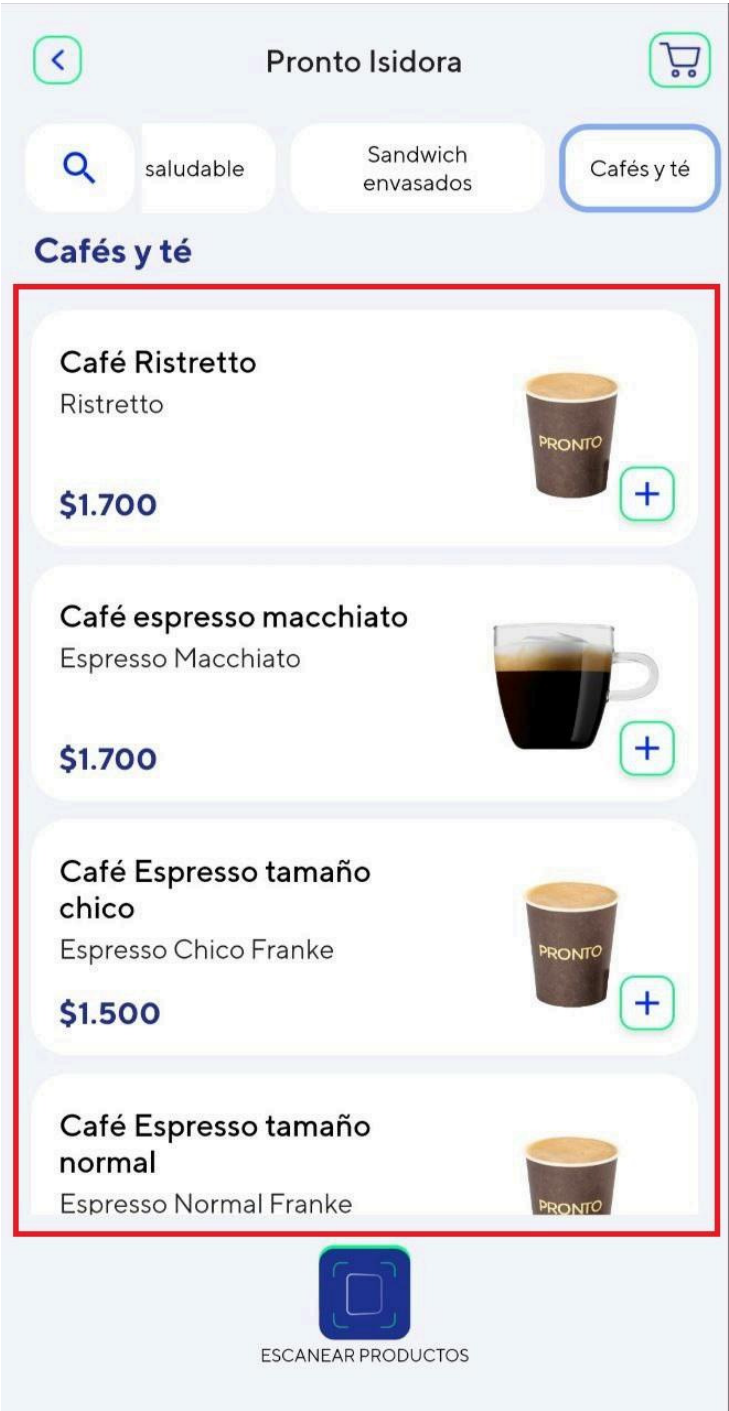


Figura 34: Cambios en la interfaz dado el orden personalizado de productos.
Fuente: Elaboración Propia.

Para el caso de los productos sustitutos y complementarios, se espera que en la pantalla de selección del producto puedan aparecer las recomendaciones abajo de la descripción del producto o en otra vista en formato *pop-up*. En la figura 35 se puede ver la interfaz de selección de producto y el espacio en el cual se espera que podrían aparecer las recomendaciones de productos sustitutos y complementarios.

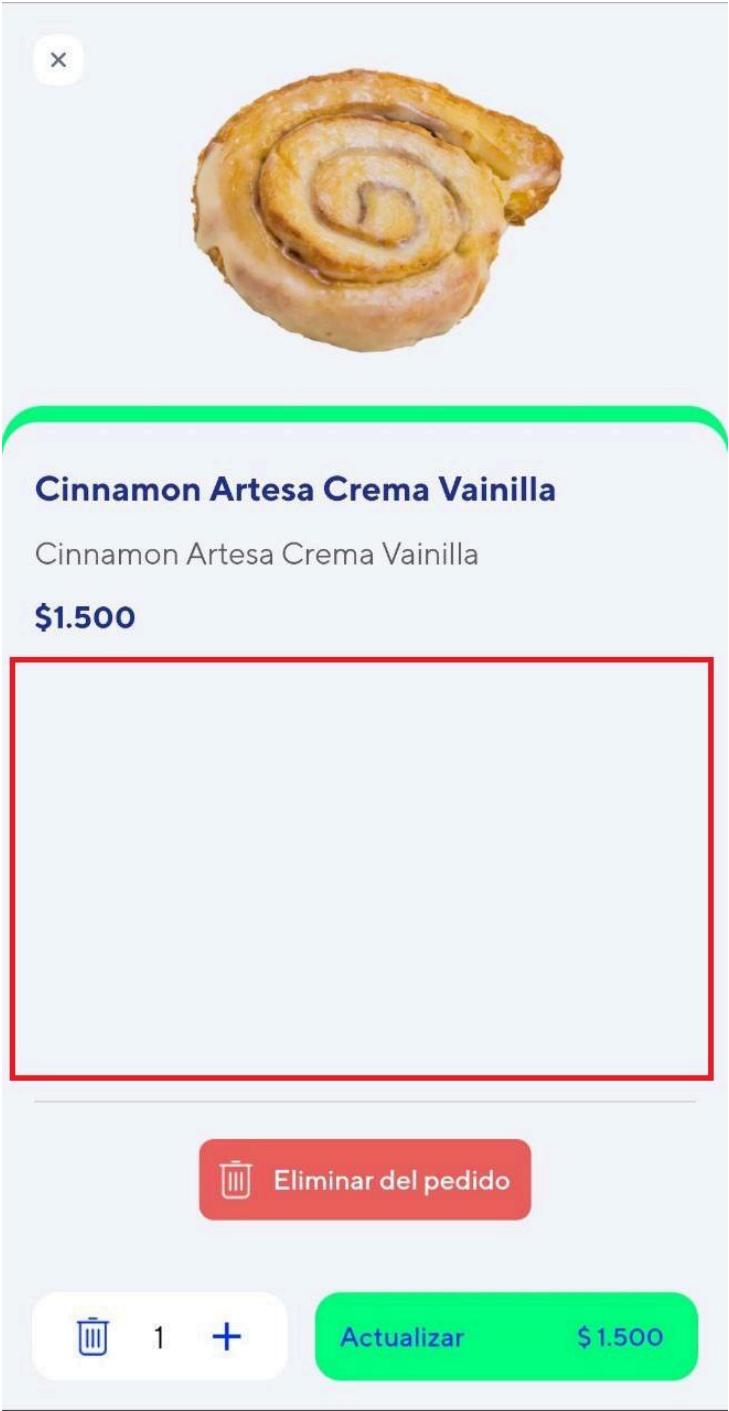


Figura 35: Cambios en la interfaz para recomendaciones de productos sustitutos y complementarios.
Fuente: Elaboración Propia.

Finalmente, para el caso de las recomendaciones personalizadas y las de usuarios similares, se espera poder incorporar las recomendaciones en el *checkout* o incorporar una sección para ellas en la vista del catálogo de Escanea & Paga. En la figura 36 se puede observar la interfaz del *check-out* y el espacio en la cual se podrían incorporar las recomendaciones.

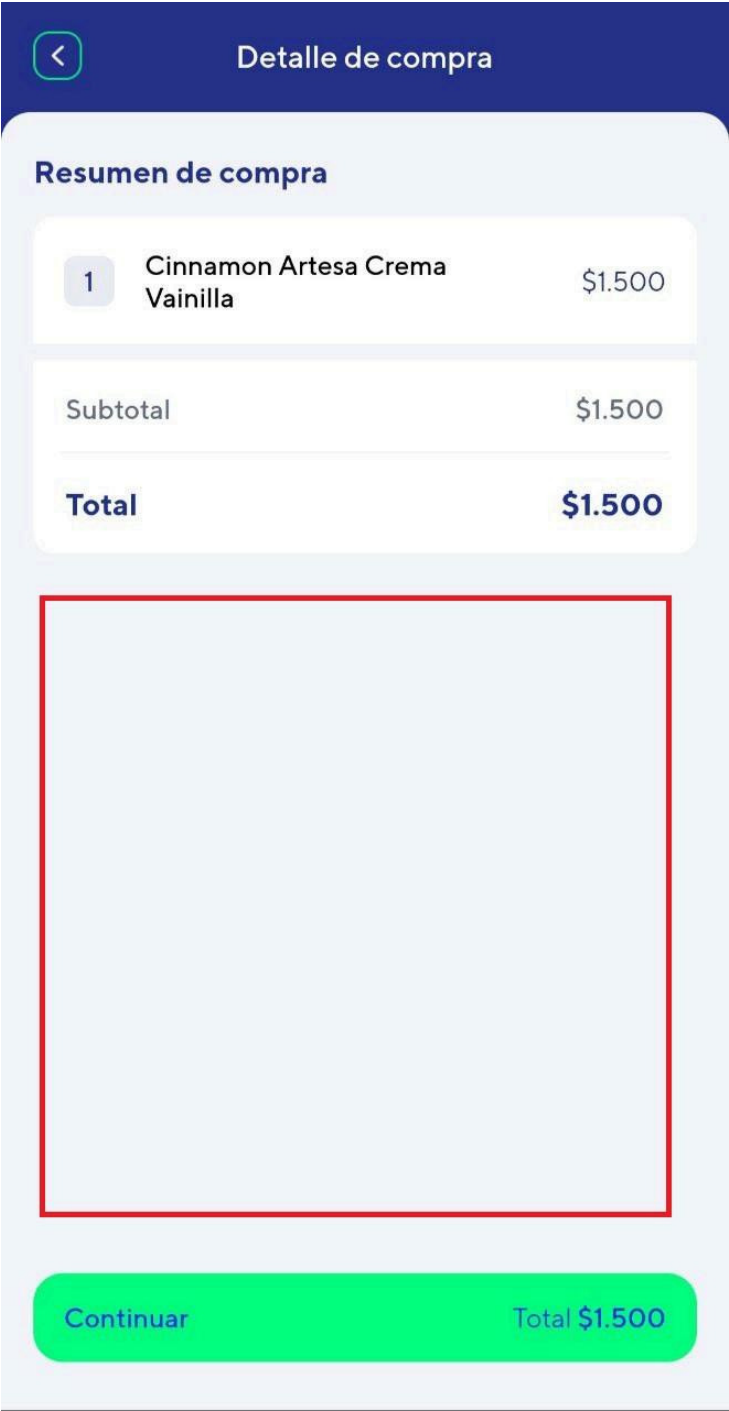


Figura 36: Cambios en la interfaz para recomendaciones personalizadas y de usuarios similares.
Fuente: Elaboración Propia.

CONCLUSIONES

En la presente sección se abordan los aprendizajes adquiridos a lo largo del desarrollo de este proyecto, el impacto que se puede llegar a provocar en caso de su implementación y las conclusiones finales de este proyecto.

Posible impacto en Copec

En el caso de que el sistema recomendador desarrollado sea implementado dentro de la aplicación Muevo, se espera que este pueda llegar a impactar los indicadores de Copec. En primera instancia, ordenar las categorías y productos personalmente va a ayudar a visibilizar productos que se ubican más abajo en el catálogo, de forma en que pueden llegar a la vista de su respectivo público de interés, por lo que si la personalización es de agrado del usuario, puede aumentar la proporción de clicks para el producto específico y en caso de que este lo compre, puede llegar a impactar la tasa de conversión del producto en particular. Los indicadores mencionados previamente impactan a nivel de producto; sin embargo, en el caso de que las recomendaciones sean de agrado de múltiples usuarios, se puede llegar a impactar indicadores a nivel general como lo son el ticket promedio y el promedio de productos por boleta. Finalmente, si la experiencia es lo suficientemente grata para los usuarios y estos llegan a fidelizarse con Muevo y Copec, esto puede llegar a impactar los grandes indicadores como lo son la cantidad de usuarios activos al mes y el *share* de pago en tiendas Pronto.

Dificultades en el desarrollo del proyecto

A lo largo del desarrollo del proyecto, existieron múltiples contratiempos que dificultaron el progreso de este. Uno de estos problemas fueron las limitaciones por parte de la memoria RAM del equipo en el cual se desarrolló el proyecto, ya que al hacer la extracción de datos y al entrenar los modelos, se utilizaba toda la memoria del equipo, donde los procesos se vieron interrumpidos y no lograban darse por finalizados. Para superar este contratiempo, se filtraron los datos para trabajar con menos transacciones y se utilizó la biblioteca *scipy* para poder comprimir las matrices *sparse*, de forma en que estas ocupen menos espacio en la memoria del equipo.

Otra dificultad relacionada a las capacidades del equipo con el que se desarrolló el proyecto fueron los altos tiempos de ejecución en Python. Realizar consultas y entrenar modelos consumió bastante tiempo, por lo que a la espera de que finalicen las ejecuciones, el avance del proyecto se vió estancado en múltiples ocasiones. Para solucionarlo, se utilizó *Amazon SageMaker*, el cual es un servicio de aprendizaje automático de *Amazon Web Services* que

provee distintos tipos de instancias, en las cuales varía su memoria y su procesador virtual, de forma que se utilizó una instancia con mayor memoria a la que posee el equipo en el cual se desarrolló el proyecto.

La última dificultad detectada durante el desarrollo del proyecto fue la poca estandarización de los datos, ya que existen múltiples productos repetidos con distintos nombres en el maestro de productos del *datalake*, códigos de productos referenciando a más de un producto y productos sin clasificación de grupo, familia o categoría. La falta de estandarización complicó el proceso del análisis exploratorio de datos ya que no se pudieron cuantificar correctamente los productos por contexto. Para solucionar esto se utilizó el maestro de productos provisto por la *stakeholder* del proyecto, sin embargo, este maestro también poseía inconsistencias en la forma en que almacena sus productos, ya que también existen productos repetidos y productos sin clasificación. El hecho de que existan productos sin clasificación dificulta la selección de productos sustitutos y complementarios, ya que el producto no pertenece a ninguna categoría, no puede ser ni sustituto ni complementario para ningún producto.

Trabajo a futuro

Como se mencionó en el capítulo anterior, este proyecto fue una prueba de concepto, por lo cual el levantamiento del sistema recomendador en la aplicación no es certero. Es por ello, que como trabajo a futuro, se debe hacer la comparación del sistema recomendador con el motor de recomendaciones de *Amazon Personalize*. Sin embargo, en caso de que el sistema recomendador desarrollado tenga un mejor desempeño que el de *Amazon Personalize*, se debe llegar a desarrollar la arquitectura de las recomendaciones en la aplicación. Además, se deberá cubrir como manejar las recomendaciones en base al stock en tiendas, ya que actualmente no se cuenta con esa información.

Por otra parte, en caso de que se busque mejorar el rendimiento del sistema recomendador desarrollado, es posible hacer una optimización de hiperparámetros, ya que fue una tarea que no se alcanzó a realizar por los contratiempos mencionados previamente y la extensión del proyecto.

Actualmente se busca implementar un motor de personalización que no se limite solamente al *E-Commerce* de Muevo, sino que interactúe con toda la aplicación. En caso de continuar con el sistema recomendador desarrollado, es posible investigar como se almacenan los combos de promociones del carrusel de la página de inicio de Muevo y los productos del catálogo de Full Copec, de forma en que se puedan relacionar estos productos con los que trabaja el sistema recomendador, para que así se puedan personalizar las secciones mencionadas con un solo sistema recomendador.

Aporte de la carrera

El procesamiento y entendimiento de datos fue un factor clave para el desarrollo del sistema recomendador, por lo que el hecho de recibir una formación teórica y técnica en esta área fue crucial para llevar a cabo este proyecto. En primera instancia, el ramo de Programación estableció una base sólida que formó el pensamiento algorítmico necesario para poder resolver problemas en base al desarrollo de algoritmos. Por otra parte, en la fase inicial del proyecto fue necesario hacer múltiples consultas para extraer los datos con los cuales se entrenaría el sistema recomendador, por lo que la asignatura de Base de Datos fue fundamental para lograr desempeñar esta labor, ya que entregó las herramientas necesarias para poder hacer consultas en bases de datos relacionales. Posteriormente, se tuvo que hacer un procesamiento para poder llevar los datos al formato requerido por los modelos, por lo que el ramo de Estadística Computacional fue indispensable, ya que a través de sus laboratorios, fue capaz de enseñar a manejar la biblioteca *Pandas*, la cual fue utilizada para llevar los datos a un formato de *dataframes*, donde luego fueron filtrados y transformados para llevarlos al formato solicitado por los modelos. Por último, para entrenar los modelos fue necesario hacer una factorización matricial, donde si bien se hizo uso de una biblioteca que facilitó la implementación, sí fue necesario entender cómo funciona el algoritmo que realiza la descomposición de las matrices, por lo que la asignatura de Computación Científica fue relevante debido a que entregó una base teórica para entender como funcionan los algoritmos de factorización matricial. Si bien existen múltiples áreas de estudio en la malla académica, fueron aquellas asignaturas relacionadas al manejo de datos y al pensamiento matemático las cuales hicieron posible la realización y culminación de este proyecto.

Conclusiones Finales

A pesar de que existieron múltiples inconvenientes con los datos para entrenar los modelos, como lo fue la falta de *feedback* explícito y la poca estandarización de los datos, fue posible desarrollar un sistema recomendador capaz de identificar los intereses de los usuarios de prueba, donde la mayoría de las recomendaciones generadas llegaron a ser del agrado de su respectivo usuario objetivo. La implementación del sistema recomendador en Muevo no busca simplemente presentar productos, sino que su real objetivo es mejorar la experiencia del usuario en la aplicación, de forma que este se sienta a gusto al utilizarla y tienda a recurrir a ella. Como la prueba en usuarios reales se completó de manera exitosa, se espera que el sistema recomendador al ser implementado en la aplicación, sea capaz de cumplir con los objetivos esperados por el negocio y provoque el impacto deseado.

ANEXOS

```
1 productos_test = '''with
2 ult_trx_por_cliente as (
3     select
4         cabecera.id_usuario,
5         cabecera.fecha_tx,
6         cabecera.codigo_empresa_codigo_tx,
7         row_number() OVER (PARTITION BY cabecera.id_usuario ORDER BY
8 cabecera.fecha_tx DESC) AS rnk
9     from copec_fidelidad_prd.trx_cabecera_rl as cabecera
10    inner join copec_fidelidad_prd.trx_detalle_rl as detalle
11        on (cabecera.codigo_empresa_codigo_tx = detalle.
12 codigo_empresa_codigo_tx)
13    where cabecera.id_usuario != 'PERSONAS_0' and
14          date(cabecera.fecha_tx) >= date '2022-01-01' and
15          date(cabecera.fecha_tx) < date '2022-11-08' and
16          cabecera.codigo_sitio in {tiendas}
17    group by 1,2,3
18 ),
19 trx_por_producto as(
20     select
21         -- count(distinct cabecera.codigo_empresa_codigo_tx) as
22         n_trx_producto,
23         -- sum(detalle.cantidad) as cantidad_productos,
24         cabecera.id_usuario,
25         detalle.codigo_producto
26     from copec_fidelidad_prd.trx_cabecera_rl as cabecera
27     inner join copec_fidelidad_prd.trx_detalle_rl as detalle
28         on (cabecera.codigo_empresa_codigo_tx = detalle.
29 codigo_empresa_codigo_tx)
30     inner join (select codigo_empresa_codigo_tx, rnk from
31 ult_trx_por_cliente where rnk = 1) as sin_ultima_trx
32         on cabecera.codigo_empresa_codigo_tx = sin_ultima_trx.
33 codigo_empresa_codigo_tx
34     where cabecera.codigo_empresa = 'PRONTO' and
35           cabecera.id_usuario != 'PERSONAS_0' and
36           date(cabecera.fecha_tx) >= date '2022-01-01' and
37           date(cabecera.fecha_tx) < date '2022-11-08' and
38           cabecera.codigo_sitio in {tiendas}
39 ), total_boletas as(
40     select
41         count(distinct cabecera.codigo_empresa_codigo_tx) as
42         total_trx_usuario,
43         cabecera.id_usuario
44     from copec_fidelidad_prd.trx_cabecera_rl as cabecera
45     inner join copec_fidelidad_prd.trx_detalle_rl as detalle
46         on (cabecera.codigo_empresa_codigo_tx = detalle.
47 codigo_empresa_codigo_tx)
48     where cabecera.codigo_empresa = 'PRONTO' and
49           cabecera.id_usuario != 'PERSONAS_0' and
```

```
42         date(cabecera.fecha_tx) >= date '2022-01-01' and
43         date(cabecera.fecha_tx) < date '2022-11-08' and
44         cabecera.codigo_sitio in {tiendas}
45     group by 2
46 )
47 select
48     a.id_usuario,
49     a.codigo_producto
50 from trx_por_producto as a
51 left join total_boletas as b
52     on (a.id_usuario = b.id_usuario)
53 where b.total_trx_usuario > 10
54     '''
55
56 productos_test = productos_test.format(tiendas = tiendas_tupla)
57
58 productos_test = wr.athena.read_sql_query(
59     sql = productos_test,
60     database = 'copec_fidelidad_prd',
61     ctas_approach=False,
62     s3_output = 's3://copec-gobierno-athena-queries-prd/copec/marketing/
practicantes/macarena_oyarzun')
```

Listing 1: Consulta última compra de usuarios con más de 10 transacciones

```
1 rating_train_usuarios_test = '''with
2 ult_trx_por_cliente as (
3     select
4         cabecera.id_usuario,
5         cabecera.fecha_tx,
6         cabecera.codigo_empresa_codigo_tx,
7         row_number() OVER (PARTITION BY cabecera.id_usuario ORDER BY
cabecera.fecha_tx DESC) AS rnk
8     from copec_fidelidad_prd.trx_cabecera_rl as cabecera
9     inner join copec_fidelidad_prd.trx_detalle_rl as detalle
10         on (cabecera.codigo_empresa_codigo_tx = detalle.
codigo_empresa_codigo_tx)
11     where cabecera.id_usuario != 'PERSONAS_0' and
12         date(cabecera.fecha_tx) >= date '2022-01-01' and
13         date(cabecera.fecha_tx) < date '2022-11-08' and
14         cabecera.codigo_sitio in {tiendas}
15     group by 1,2,3
16 ),
17 trx_por_producto as(
18     select
19         count(distinct cabecera.codigo_empresa_codigo_tx) as
n_trx_producto,
20         sum(detalle.cantidad) as cantidad_productos,
21         cabecera.id_usuario,
22         detalle.codigo_producto
23     from copec_fidelidad_prd.trx_cabecera_rl as cabecera
24     inner join copec_fidelidad_prd.trx_detalle_rl as detalle
25         on (cabecera.codigo_empresa_codigo_tx = detalle.
```

```

26     codigo_empresa_codigo_tx)
27     inner join (select codigo_empresa_codigo_tx, rnk from
ult_trx_por_cliente where rnk > 1) as sin_ultima_trx
28     on cabecera.codigo_empresa_codigo_tx = sin_ultima_trx.
codigo_empresa_codigo_tx
29     where cabecera.codigo_empresa = 'PRONTO' and
cabecera.id_usuario != 'PERSONAS_0' and
30     date(cabecera.fecha_tx) >= date '2022-01-01' and
31     date(cabecera.fecha_tx) < date '2022-11-08' and
32     cabecera.codigo_sitio in {tiendas}
33     group by 3,4
34 ), total_boletas as(
35     select
36         count(distinct cabecera.codigo_empresa_codigo_tx) as
total_trx_usuario,
37         cabecera.id_usuario
38     from copec_fidelidad_prd.trx_cabecera_rl as cabecera
39     inner join copec_fidelidad_prd.trx_detalle_rl as detalle
40     on (cabecera.codigo_empresa_codigo_tx = detalle.
codigo_empresa_codigo_tx)
41     inner join (select codigo_empresa_codigo_tx from ult_trx_por_cliente
where rnk > 1) as sin_ultima_trx
42     on cabecera.codigo_empresa_codigo_tx = sin_ultima_trx.
codigo_empresa_codigo_tx
43     where cabecera.codigo_empresa = 'PRONTO' and
cabecera.id_usuario != 'PERSONAS_0' and
44     date(cabecera.fecha_tx) >= date '2022-01-01' and
45     date(cabecera.fecha_tx) < date '2022-11-08' and
46     cabecera.codigo_sitio in {tiendas}
47     group by 2
48 )
49 )
50 select
51     a.id_usuario,
52     a.codigo_producto,
53     a.n_trx_producto,
54     b.total_trx_usuario
55 from trx_por_producto as a
56 left join total_boletas as b
57     on (a.id_usuario = b.id_usuario)
58 where total_trx_usuario >= 10
59     '''
60
61 rating_train_usuarios_test = rating_train_usuarios_test.format(tiendas =
tiendas_tupla)
62
63 rating_train_usuarios_test = wr.athena.read_sql_query(
64     sql = rating_train_usuarios_test,
65     database = 'copec_fidelidad_prd',
66     ctas_approach=False,
67     s3_output = 's3://copec-gobierno-athena-queries-prd/copec/marketing/

```

practicantes/macarena_oyarzun')

Listing 2: Consulta productos comprados por usuarios con más de 10 transacciones (Sin incluir la última transacción)

```
1 rating_bajo_10 = '''with
2 ult_trx_por_cliente as (
3     select
4         cabecera.id_usuario,
5         cabecera.fecha_tx,
6         cabecera.codigo_empresa_codigo_tx,
7         row_number() OVER (PARTITION BY cabecera.id_usuario ORDER BY
8 cabecera.fecha_tx DESC) AS rnk
9     from copec_fidelidad_prd.trx_cabecera_rl as cabecera
10    inner join copec_fidelidad_prd.trx_detalle_rl as detalle
11        on (cabecera.codigo_empresa_codigo_tx = detalle.
12 codigo_empresa_codigo_tx)
13    where cabecera.id_usuario != 'PERSONAS_0' and
14          date(cabecera.fecha_tx) >= date '2022-01-01' and
15          date(cabecera.fecha_tx) < date '2022-11-08' and
16          cabecera.codigo_sitio in {tiendas}
17    group by 1,2,3
18 ),
19 trx_por_producto as(
20     select
21         count(distinct cabecera.codigo_empresa_codigo_tx) as
22 n_trx_producto,
23         sum(detalle.cantidad) as cantidad_productos,
24         cabecera.id_usuario,
25         detalle.codigo_producto
26     from copec_fidelidad_prd.trx_cabecera_rl as cabecera
27    inner join copec_fidelidad_prd.trx_detalle_rl as detalle
28        on (cabecera.codigo_empresa_codigo_tx = detalle.
29 codigo_empresa_codigo_tx)
30    where cabecera.codigo_empresa = 'PRONTO' and
31          cabecera.id_usuario != 'PERSONAS_0' and
32          date(cabecera.fecha_tx) >= date '2022-01-01' and
33          date(cabecera.fecha_tx) < date '2022-11-08' and
34          cabecera.codigo_sitio in {tiendas}
35    group by 3,4
36 ), total_boletas as(
37     select
38         count(distinct cabecera.codigo_empresa_codigo_tx) as
39 total_trx_usuario,
40         cabecera.id_usuario
41     from copec_fidelidad_prd.trx_cabecera_rl as cabecera
42    inner join copec_fidelidad_prd.trx_detalle_rl as detalle
43        on (cabecera.codigo_empresa_codigo_tx = detalle.
44 codigo_empresa_codigo_tx)
45    where cabecera.codigo_empresa = 'PRONTO' and
46          cabecera.id_usuario != 'PERSONAS_0' and
47          date(cabecera.fecha_tx) >= date '2022-01-01' and
48          date(cabecera.fecha_tx) < date '2022-11-08' and
```

```
43         cabecera.codigo_sitio in {tiendas}
44     group by 2
45 )
46 select
47     a.id_usuario,
48     a.codigo_producto,
49     a.n_trx_producto,
50     b.total_trx_usuario
51 from trx_por_producto as a
52 left join total_boletas as b
53     on (a.id_usuario = b.id_usuario)
54 where total_trx_usuario <= 10
55     '''
56
57 rating_bajo_10 = rating_bajo_10.format(tiendas = tiendas_tupla)
58
59 rating_bajo_10 = wr.athena.read_sql_query(
60     sql = rating_bajo_10,
61     database = 'copec_fidelidad_prd',
62     ctas_approach=False,
63     s3_output = 's3://copec-gobierno-athena-queries-prd/copec/marketing/
practicantes/macarena_oyarzun')
```

Listing 3: Consulta productos comprados por usuarios con 10 transacciones o menos

```
1 rating_test_completo = '''with
2 ult_trx_por_cliente as (
3     select
4         cabecera.id_usuario,
5         cabecera.fecha_tx,
6         cabecera.codigo_empresa_codigo_tx,
7         row_number() OVER (PARTITION BY cabecera.id_usuario ORDER BY
cabecera.fecha_tx DESC) AS rnk
8     from copec_fidelidad_prd.trx_cabecera_rl as cabecera
9     inner join copec_fidelidad_prd.trx_detalle_rl as detalle
10         on (cabecera.codigo_empresa_codigo_tx = detalle.
codigo_empresa_codigo_tx)
11     where cabecera.id_usuario != 'PERSONAS_0' and
12         date(cabecera.fecha_tx) >= date '2022-01-01' and
13         date(cabecera.fecha_tx) < date '2022-11-08' and
14         cabecera.codigo_sitio in {tiendas}
15     group by 1,2,3
16 ),
17 trx_por_producto as(
18     select
19         count(distinct cabecera.codigo_empresa_codigo_tx) as
n_trx_producto,
20         sum(detalle.cantidad) as cantidad_productos,
21         cabecera.id_usuario,
22         detalle.codigo_producto
23     from copec_fidelidad_prd.trx_cabecera_rl as cabecera
24     inner join copec_fidelidad_prd.trx_detalle_rl as detalle
25         on (cabecera.codigo_empresa_codigo_tx = detalle.
```

```

    codigo_empresa_codigo_tx)
26     where cabecera.codigo_empresa = 'PRONTO' and
27           cabecera.id_usuario != 'PERSONAS_0' and
28           date(cabecera.fecha_tx) >= date '2022-01-01' and
29           date(cabecera.fecha_tx) < date '2022-11-08' and
30           cabecera.codigo_sitio in {tiendas}
31     group by 3,4
32 ), total_boletas as(
33     select
34         count(distinct cabecera.codigo_empresa_codigo_tx) as
total_trx_usuario,
35         cabecera.id_usuario
36     from copec_fidelidad_prd.trx_cabecera_rl as cabecera
37     inner join copec_fidelidad_prd.trx_detalle_rl as detalle
38         on (cabecera.codigo_empresa_codigo_tx = detalle.
codigo_empresa_codigo_tx)
39     where cabecera.codigo_empresa = 'PRONTO' and
40           cabecera.id_usuario != 'PERSONAS_0' and
41           date(cabecera.fecha_tx) >= date '2022-01-01' and
42           date(cabecera.fecha_tx) < date '2022-11-08' and
43           cabecera.codigo_sitio in {tiendas}
44     group by 2
45 )
46 select
47     a.id_usuario,
48     a.codigo_producto,
49     a.n_trx_producto,
50     b.total_trx_usuario
51 from trx_por_producto as a
52 left join total_boletas as b
53     on (a.id_usuario = b.id_usuario)
54 where total_trx_usuario > 10
55     '''
56
57 rating_test_completo = rating_test_completo.format(tiendas =
tiendas_tupla)
58
59 rating_test_completo = wr.athena.read_sql_query(
60     sql = rating_test_completo,
61     database = 'copec_fidelidad_prd',
62     ctas_approach=False,
63     s3_output = 's3://copec-gobierno-athena-queries-prd/copec/marketing/
practicantes/macarena_oyarzun')
```

Listing 4: Consulta productos comprados por usuarios con más de 10 transacciones
(Incluyendo los productos de la última transacción)

REFERENCIAS BIBLIOGRÁFICAS

- [pyt, 1991] (1991). Welcome to python.org. <https://www.python.org/>. Último acceso: 23/03/2023.
- [jup, 2014] (2014). Project jupyter. <https://www.python.org/>. Último acceso: 23/03/2023.
- [sag, 2017] (2017). Amazon sagemaker. <https://aws.amazon.com/es/sagemaker/>. Último acceso: 23/03/2023.
- [Bell y Koren, 2007] Bell, R. M. y Koren, Y. (2007). Scalable collaborative filtering with jointly derived neighborhood interpolation weights. *Seventh IEEE International Conference on Data Mining (ICDM 2007)*.
- [Chapman et al., 2000] Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., y Wirth, R. (2000). *CRISP-DM 1.0: Step-by-step data mining guide*. SPSS.
- [Copec, 1970] Copec (1970). Pasado, presente y futuro, copec cumplió 86 años. <https://web.copec.cl/nuestra-empresa/historia>. Último acceso: 02/01/2023.
- [Fayyad et al., 1996] Fayyad, U., Piatetsky-Shapiro, G., y Smyth, P. (1996). The kdd process for extracting useful knowledge from volumes of data. *Communications of the ACM*, 39(11):27-34.
- [Hu et al., 2008] Hu, Y., Koren, Y., y Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. *2008 Eighth IEEE International Conference on Data Mining*.
- [Jensen, 2012] Jensen, K. (2012). File:crisp-dm process diagram.png. https://commons.wikimedia.org/w/index.php?title=File%3ACRISP-DM_Process_Diagram.png&oldid=506972775. Último acceso: 19/02/2023.
- [Johnson, 2014] Johnson, C. C. (2014). Logistic matrix factorization for implicit feedback data. <https://stanford.edu/~rezab/nips2014workshop/submits/logmat.pdf>. Último acceso: 20/03/2023.
- [Koren et al., 2009] Koren, Y., Bell, R., y Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8):30-37.
- [Li, 2017] Li, S. (2017). A gentle introduction on market basket analysis - association rules. <https://towardsdatascience.com/a-gentle-introduction-on-market-basket-analysis-association-rules-fa4b986a40ce>. Último acceso: 23/03/2023.
- [Martinez-Plumed et al., 2021] Martinez-Plumed, F., Contreras-Ochando, L., Ferri, C., Hernandez-Orallo, J., Kull, M., Lachiche, N., Ramirez-Quintana, M. J., y Flach, P. (2021).

Crisp-dm twenty years later: From data mining processes to data science trajectories. *IEEE Transactions on Knowledge and Data Engineering*, 33(8):3048–3061.

[Muevo, 2020] Muevo (2020). Descubre los beneficios: Muevo copec. <https://www.muevocopec.cl/>. Último acceso: 02/01/2023.

[Peralta, 2014] Peralta, F. C. (2014). Proceso de conceptualización del entendimiento del negocio para proyectos de explotación de información. *Revista Latinoamericana de Ingeniería de Software*, 2(5):273.

[Punto, 2001] Punto (2001). Punto copec - copec s.a.: Primera en servicio. <https://ww2.copec.cl/personas/punto>. Último acceso: 02/01/2023.

[Ricci et al., 2022] Ricci, F., Rokach, L., y Shapira, B. (2022). *Recommender Systems Handbook*. Springer.

[SEMMA, 2012] SEMMA, S. . (2012). Sas enterprise miner. <https://web.archive.org/web/20120308165638/http://www.sas.com/offices/europe/uk/technologies/analytics/datamining/miner/semma.html>. Último acceso: 02/01/2023.