Facultad de Ciencias Económicas y Empresariales

ICADE

Análisis de la cesta de la compra en Instacart: extracción de patrones de consumo implementando el algoritmo *a priori*.

Autora: Lucía Gema Hernando García

Directora: Anitha Srinivasan

Madrid | Febrero 2024

Resumen

En el presente trabajo se llevará a cabo un análisis de la cesta de la compra en base a los pedidos realizados por diversos usuarios a través de la plataforma digital Instacart. El propósito principal de esta investigación consiste en identificar patrones de compra y relaciones entre productos con el fin de optimizar estrategias comerciales y mejorar la experiencia del usuario en la compra *online*. A partir de un análisis exhaustivo de los datos y la implementación del algoritmo *a priori*, se ha descubierto una preferencia creciente por productos ecológicos y saludables entre los consumidores, con una alta frecuencia de compra de productos frescos, y un mayor nivel de actividad los fines de semana. Como contribución a esta investigación, se ofrece un marco teórico que abarca las técnicas implementadas, así como un análisis de la evolución y el modelo de negocio de Instacart.

Palabras clave: minería de datos, reglas de asociación, conjuntos frecuentes, soporte, confianza, algoritmo *a priori*, Instacart, cesta de la compra, patrones de compra, comportamiento del consumidor.

Abstract

The present study will conduct a market basket analysis based on orders made by various users through the digital platform Instacart. The main purpose of this research is to identify purchasing patterns and relationships between products in order to optimize commercial strategies and enhance the user experience in online shopping. Through a comprehensive analysis of the data and the implementation of the *a priori* algorithm, a growing preference for eco-friendly and healthy products among consumers has been discovered, with a high frequency of purchasing fresh products and increased activity levels during weekends. As a contribution to this research, a theoretical framework covering the implemented techniques is provided, along with an analysis of the evolution and business model of Instacart.

Keywords: data mining, association rules, frequent itemsets, support, confidence, *a priori* algorithm, Instacart, market basket analysis, purchase patterns, consumer behavior.

Índice de contenidos

Índice de figuras

Índice de tablas

1. Introducción.

En un mundo cada vez más digitalizado, el comercio electrónico ha experimentado un crecimiento explosivo. La pandemia del COVID-19 ha acelerado aún más esta tendencia, alcanzando un aumento significativo en las compras *online* a través de plataformas digitales.

Este cambio de paradigma en el comportamiento del consumidor ha dado lugar a una riqueza de datos sin precedentes sobre los hábitos y preferencias de los consumidores. Desde la exploración de productos hasta la toma de decisiones de compra, cada interacción deja un rastro digital valioso que puede traducirse en conocimientos profundos sobre las motivaciones del consumidor (Chaudhuri et al., 2021).

La minería de datos, una técnica poderosa para extraer patrones y reglas significativas a partir de grandes conjuntos de datos, se ha convertido en un pilar fundamental en la comprensión de estos comportamientos (Moya Amaris y Rodríguez Rodríguez, 2003).

Por ejemplo, empresas líderes en el comercio electrónico como Amazon utilizan la minería de datos para analizar el historial de compras de sus clientes y recomendar productos de manera personalizada. Estas recomendaciones, basadas en algoritmos avanzados que tienen en cuenta factores como preferencias pasadas, tendencias de compra y perfiles de usuario, mejoran significativamente la experiencia del cliente (Isinkaye et al., 2015).

Además, el conocimiento de patrones de compra de los usuarios en las plataformas digitales permite implementar estrategias comerciales innovadoras, como por ejemplo estrategias basadas en precios dinámicos. Empresas como Uber ajustan los precios en tiempo real según la demanda y otros factores, maximizando así sus ingresos y optimizando la utilización de sus servicios (Lunde y JIANG, 2023).

Sin embargo, la minería de datos no solo se involucra en el entendimiento y la búsqueda de patrones de comportamiento, sino que abarca una amplia gama de campos de aplicación, incluyendo el comercio y la banca. En estos sectores, esta técnica se emplea para llevar a cabo segmentaciones de clientes, previsiones de ventas y análisis de riesgos. Además, en el ámbito de la medicina y la farmacología se utiliza para el diagnóstico de enfermedades y para evaluar la efectividad de los tratamientos.

También, en disciplinas como la astronomía, la minería de datos se aplica en la identificación de nuevas galaxias y estrellas, mientras que en geología, minería, agricultura y pesca, se utiliza para identificar áreas óptimas para cultivos, pesca o explotación minera a partir de bases de datos de imágenes de satélites.

Por otro lado, en ciencias ambientales, esta técnica ayuda a identificar modelos de funcionamiento de ecosistemas naturales y artificiales con el propósito de mejorar su observación, gestión y control. Otro campo de aplicación importante es la seguridad y la detección de fraude, donde se emplean técnicas como el reconocimiento facial e identificaciones biométricas (Riquelme Santos et al., 2006).

1.1. Objetivos y metodología.

Este trabajo se centra en el análisis de las compras realizadas a través de la plataforma digital Instacart. Dicha plataforma ha ganado popularidad en Estados Unidos al ofrecer a los consumidores una forma conveniente y eficiente de adquirir productos de supermercado desde la comodidad de sus hogares.

El valor que se pretende aportar con esta investigación radica en la identificación de tendencias y patrones de compra que puedan ser utilizados por Instacart para optimizar la experiencia del usuario y, en consecuencia, atraer a un mayor número de consumidores. Específicamente, se desean lograr los siguientes **objetivos**:

- Alcanzar una mayor comprensión acerca de las transacciones registradas en la base de datos de Instacart.
- Identificar productos que tengan una frecuencia de compra significativamente mayor que otros en la búsqueda de patrones.
- Descubrir relaciones entre los productos comprados, asegurando una alta fiabilidad en las asociaciones encontradas.
- Ampliar el análisis de los productos más frecuentes incorporando variables como los días de la semana en los que se compran, las veces que se vuelven a comprar, o los departamentos a los que pertenecen, entre otras.
- Obtener *insights* sobre el comportamiento de los consumidores y posibles estrategias comerciales.

Para alcanzar estos objetivos, se ha empleado una **metodología** basada en el análisis exploratorio del conjunto de datos y la implementación del algoritmo *a priori*, utilizando *R Studio* como herramienta principal. Sin embargo, como en toda metodología, se han identificado algunas desventajas. En este caso, se ha encontrado una gran limitación computacional debido a la extensa base de datos y la complejidad del funcionamiento del algoritmo implementado.

Por otro lado, la investigación ha seguido un enfoque inductivo, pues a partir del análisis de los datos se han identificado interesantes relaciones entre diversos productos. Asimismo, se ha aplicado un método cuantitativo, aprovechando las técnicas de minería de datos para descubrir patrones ocultos entre las transacciones realizadas a través de Instacart.

Para llevar a cabo la investigación, se ha utilizado como fuente principal diversos conjuntos de datos proporcionados por la mencionada plataforma. Estos conjuntos contienen información detallada sobre los pedidos realizados por los usuarios de Instacart.

El trabajo se estructura en **cuatro partes**. En primer lugar, se expone un marco teórico como base para el entendimiento de la investigación. Por un lado, se explica en profundidad la técnica de minería de datos, específicamente las reglas de asociación, y la lógica del algoritmo *a priori*. Por otro lado, se detalla la evolución y el modelo de negocio de la plataforma digital Instacart.

En segundo lugar, se lleva a cabo el estudio de campo, primero realizando un análisis en profundidad de los conjuntos de datos utilizados, y después implementando el algoritmo *a priori* sobre las transacciones.

En tercer lugar, se presentan los resultados del estudio incluyendo los patrones de compra identificados y, en cuarto lugar, se ofrecen conclusiones basadas en los hallazgos del mismo.

2. Estado del arte.

2.1. Minería de datos y reglas de asociación.

La minería de datos consiste en la extracción de patrones y reglas significativas a partir de una amplia cantidad de información. Esta técnica es de gran valor en cualquier ámbito en el que existe una considerable cantidad de datos disponibles y un potencial para adquirir conocimientos a partir de ellos (Moya Amaris y Rodríguez Rodríguez, 2003).

Las técnicas de minería de datos se clasifican en dos categorías principales: supervisadas y no supervisadas. En las técnicas supervisadas, cada observación tiene asignado un valor que identifica su clase. Sin embargo, en las técnicas no supervisadas, las observaciones no tienen clases asociadas.

Algunos ejemplos de técnicas supervisadas incluyen árboles de decisión, inducción neuronal, regresión y series temporales. Por otro lado, dentro de las no supervisadas se encuentran técnicas de segmentación, agrupamiento o *clustering*, detección de desviaciones, patrones secuenciales y reglas de asociación (Beltrán Martínez, 2001). Este proyecto se enfocará en esta última técnica.

Para ilustrar la técnica de reglas de asociación, se expone un ejemplo de análisis de la cesta de la compra, comúnmente utilizado en el ámbito de comercio y marketing. Supongamos que se dispone de una base de datos que contiene información sobre transacciones realizadas por diversos clientes en un supermercado.

Por un lado, se dispone de un conjunto de artículos o *ítems* $D = \{d_1, d_2, ..., d_m\}$ donde m denota el número total de artículos y, por otro lado, de un conjunto de transacciones $T = \{t_1, t_2, ..., t_n\}$ donde n denota el número total de transacciones. Cada transacción t está formada por un conjunto de *ítems* que forman parte de D (Malberti Riveros y Elida Beguerí, 2015).

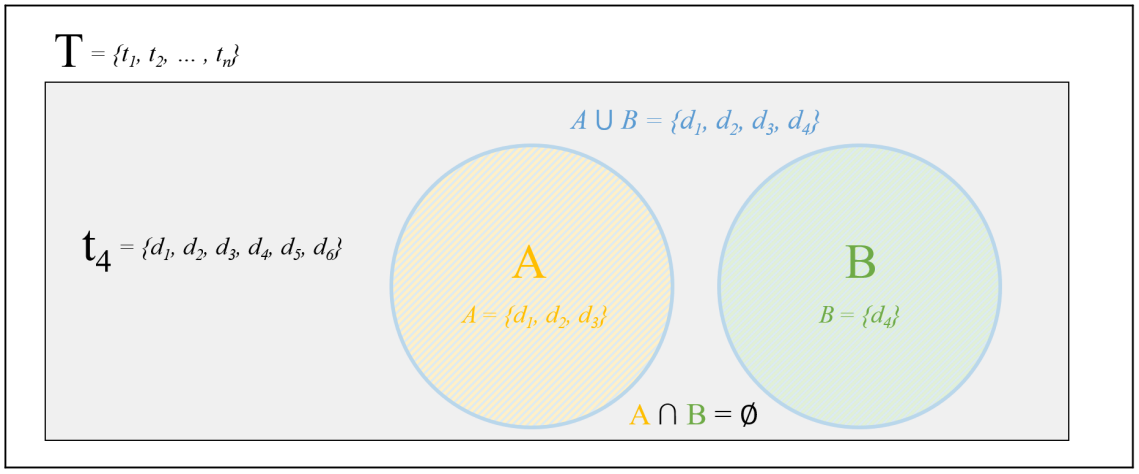
Esta técnica permite descubrir correlaciones o coocurrencias entre los *ítems*, que se formalizarán posteriormente en reglas de tipo "si…entonces…" (Beltrán Martínez, 2001). Un ejemplo sería una regla que indique que existe una alta probabilidad de que si se compra café, entonces se compren galletas.

Así pues, una regla de asociación se representa en la forma $A \rightarrow B$ e indica que cuando se compra A, es probable que se compre también B. Tanto A como B son conjuntos que pueden estar compuestos por uno o varios *ítems*.

Ahora bien, es fundamental considerar las implicaciones de una regla de asociación $A \rightarrow B$. Por un lado, los conjuntos A y B están formados por artículos que pertenecen a D y se consideran conjuntos mutuamente excluyentes. Esto significa que A y B no comparten elementos entre sí ($A \cap B = \varnothing$). Por otro lado, el conjunto $A \cup B$ debe estar contenido en alguna transacción de T o ser igual que alguna de ellas (Malberti Riveros y Elida Beguerí, 2015).

A continuación, la Figura 1 ejemplifica las implicaciones de una regla de asociación. Supóngase que $A = \{d_1, d_2, d_3\}$, $B = \{d_4\}$ y $t_4 = \{d_1, d_2, d_3, d_4, d_5, d_6\}$. Por un lado se cumple la condición $A \cap B = \varnothing$ ya que ambos conjuntos no tienen *ítems* en común. Por otro lado, el conjunto $A \cup B = \{d_1, d_2, d_3, d_4\}$ se encuentra contenido en alguna transacción de T, concretamente, en t_4.

Figura 1. Implicaciones de una regla de asociación.



Fuente: elaboración propia

Para saber si una regla de asociación es adecuada, se deben tener en cuenta dos criterios: el soporte y la confianza. El **soporte** de un *itemset* o conjunto de *ítems* se define como la probabilidad de que las transacciones (T) contengan dicho *itemset*. Por tanto, se

calcula como el número total de transacciones en las que aparece el conjunto de *ítems*, dividido entre *n* (Sáenz López et al., 2017).

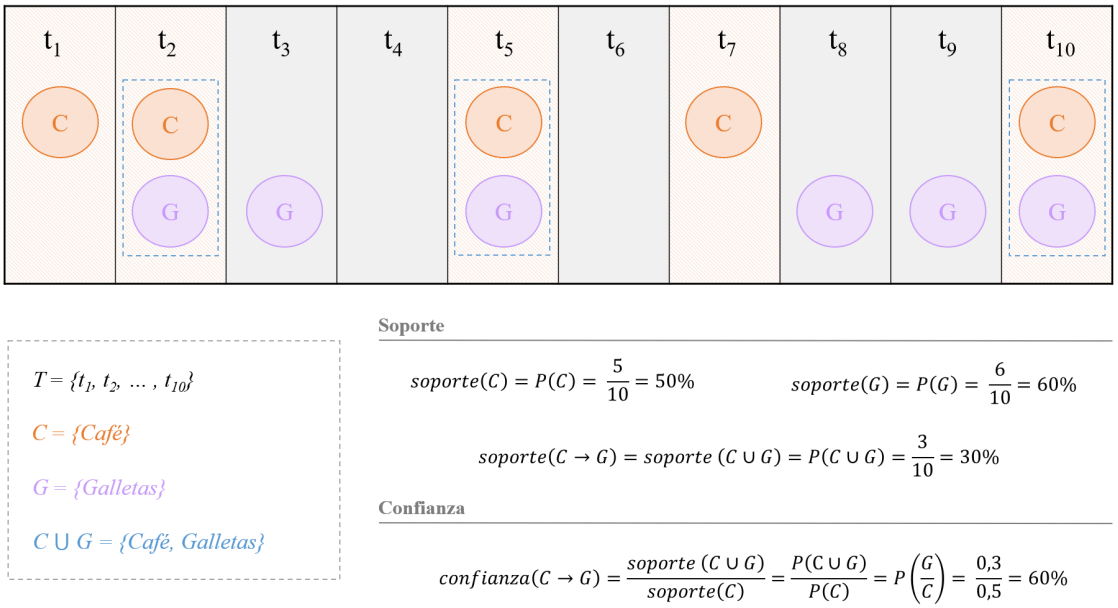
Por ejemplo, supóngase que se tienen los conjuntos $C = \{Café\}$ y $G = \{Galletas\}$. El soporte de *C* será la proporción de transacciones en las que aparece *Café*, y el soporte de *G* será la proporción de transacciones en las que aparece *Galletas*. Por lo tanto, siguiendo la Figura 2, donde *P(i)* denota la probabilidad de que aparezca un conjunto *i*, y asumiendo que el número total de transacciones es $n = 10$, el soporte de *Café* y *Galletas* es del 50% y 60%, respectivamente.

Supóngase ahora que se quiere estudiar la regla de asociación $C \rightarrow G$. Para ello, primero se debe calcular el soporte del conjunto $C \cup G$, es decir, la proporción de transacciones en las que aparecen *C* y *G* conjuntamente. Siguiendo de nuevo la Figura 2, el soporte ($C \rightarrow G$) es del 30%, lo que significa que existe una probabilidad del 30% de que los artículos *Café* y *Galletas* se compren de forma conjunta.

La **confianza** de una regla de asociación $C \rightarrow G$ indica la probabilidad de que se compre *G*, habiendo comprado *C*. En consecuencia, la confianza de dicha regla se calcula como el soporte ($C \rightarrow G$) dividido entre el soporte (*C*) (Sáenz López et al., 2017).

Siguiendo el ejemplo anterior y como muestra la Figura 2, la confianza de la regla ($C \rightarrow G$) es del 60%. Esto implica que existe una probabilidad del 60% de que se compren *Galletas* habiendo comprado *Café*. Los productos *Café* y *Galletas* aparecen juntos en el 60% de las transacciones en las que se ha comprado *Café*.

Figura 2. Ejemplo de soporte y confianza.



Fuente: elaboración propia

La Tabla 1 muestra un resumen de las expresiones matemáticas de los conceptos de soporte y confianza, previamente explicados.

Tabla 1. Soporte y confianza de una regla de asociación.

Soporte	$soporte(A) = P(A)$ $soporte(A \rightarrow B) = soporte\ (A \cup B) = P(A \cup B)$
Confianza	$confianza(A \rightarrow B) = \frac{soporte\ (A \cup B)}{soporte(A)} = \frac{P(A \cup B)}{P(A)} = P\left(\frac{B}{A}\right)$

Fuente: Malberti Riveros y Elida Beguerí, 2015

Una regla de asociación será apropiada siempre y cuando satisfaga el soporte mínimo y la confianza mínima establecida. Si un *itemset* satisface el soporte mínimo, se considerará frecuente, y si dicho *itemset* frecuente supera la confianza mínima, su correlación se formalizará en una regla de asociación (Sáenz López et al., 2017).

Siguiendo el ejemplo anterior, si el soporte mínimo es del 20%, el *itemset {Café, Galletas}* se entiende como frecuente ya que su soporte satisface el mínimo. Ahora bien, si la confianza mínima es del 30%, la regla de asociación *Café* → *Galletas* será adecuada ya que la confianza del *itemset* frecuente supera la mínima establecida.

Tras haber encontrado patrones y reglas de asociación a través del análisis de la cesta de la compra, el supermercado será capaz de diseñar estrategias de comercialización, especialmente estrategias de *merchandising*. Este último término se refiere a las técnicas enfocadas en la disposición y presentación de los artículos en el punto de venta con el objetivo de atraer clientes y generar un mayor número de ventas (Verastegui Tene y Vargas Merino, 2021).

En este caso, se ha encontrado la regla de asociación *Café* → *Galletas* que sugiere que si se compra *Café*, entonces se compran *Galletas*. Ahora, el supermercado podrá implementar una estrategia con la que, por ejemplo, decida colocar las galletas cerca del café para despertar el interés del cliente por comprarlas, aunque su intención inicial fuese únicamente comprar café.

Esta regla se caracteriza por ser booleana, pues hace referencia a la presencia o ausencia de un *ítem* en una transacción, concretamente a la presencia o ausencia de *Galletas* en una transacción que contiene *Café*. Además, se trata de una regla de dimensión simple, ya que se enfoca exclusivamente en una dimensión: la compra del *ítem*, en este caso, la compra de *Galletas* (Moya Amaris y Rodríguez Rodríguez, 2003).

2.2. Métricas para la evaluación de reglas de asociación.

Existen diversas métricas que se utilizan para la evaluación de las reglas de asociación una vez que han sido identificadas. Tanto el soporte como la confianza son medidas comunes para evaluar reglas. No obstante, también se prestará atención a otras tres: cobertura (*coverage*), interés (*lift*), y Chi-cuadrado.

En primer lugar, la **cobertura o *coverage*** mide la probabilidad de que una regla de asociación $A \rightarrow B$ pueda aplicarse a una transacción seleccionada al azar. Se calcula como la proporción de transacciones que contienen el antecedente de la regla, es decir, es el soporte del antecedente de la regla de asociación. Por ejemplo, la cobertura de la

regla $A \rightarrow B$ sería el soporte de A. La cobertura de una regla puede tomar valores entre cero y uno, donde cero indica que A no aparece en ninguna transacción y uno indica que A aparece en todas las transacciones (Hahsler, 2015).

El **interés o *lift*** de una regla de asociación $A \rightarrow B$ representa las veces que ocurren A y B de manera conjunta con respecto a las veces que ocurrirían si fuesen estadísticamente independientes. La métrica *lift* compara la confianza de la regla $A \rightarrow B$ con respecto al soporte de B y toma valores de cero a infinito. Si el *lift* toma el valor uno, significa que A y B son estadísticamente independientes y, por tanto, no existe asociación entre ambos conjuntos. Sin embargo, cuando toma valores mayores que uno, se entiende que existe asociación entre A y B, pues ocurren de forma conjunta más frecuentemente de lo esperado (Hahsler, 2015).

Finalmente, considerando una regla de asociación $A \rightarrow B$, la prueba estadística de **Chi-cuadrado** evalúa la independencia de A y B mediante el estadístico Chi-cuadrado. Este estadístico se calcula a partir de la tabla de contingencia específica para la regla en cuestión. Se trata de la suma de las diferencias al cuadrado de las frecuencias observadas (Oi) y esperadas (Ei), divididas entre las frecuencias esperadas (Ei).

El estadístico Chi-cuadrado puede tomar valores entre cero e infinito. Bajo un nivel de confianza del 95% y un grado de libertad, el valor crítico de una distribución de probabilidad Chi-cuadrado es 3,84. Si el valor del estadístico es mayor que dicho valor crítico, se debe rechazar la hipótesis nula de independencia entre A y B. Rechazar la hipótesis nula implica la existencia de una relación fuerte entre A y B. Por otro lado, un valor de cero sugiere que A y B son estadísticamente independientes, indicando la ausencia de asociación entre ambos conjuntos (Hahsler, 2015).

La Tabla 2 muestra un resumen de las expresiones matemáticas de las métricas cobertura, *lift* y estadístico Chi-cuadrado.

Tabla 2. Cobertura, *lift* y estadístico Chi-cuadrado para la evaluación de reglas.

Cobertura	$cobertura(A \rightarrow B) = soporte(A) = P(A)$
Lift	$lift(A \rightarrow B) = \frac{confianza(A \rightarrow B)}{soporte(B)} = \frac{P(A \cup B)}{P(A)P(B)}$
Estadístico Chi-cuadrado	$chi - cuadrado(A \rightarrow B) = \sum_i \frac{(O_i - E_i)^2}{E_i}$

Fuente: Hahsler, 2015

2.3. Aproximación a la herramienta utilizada: algoritmo *a priori*.

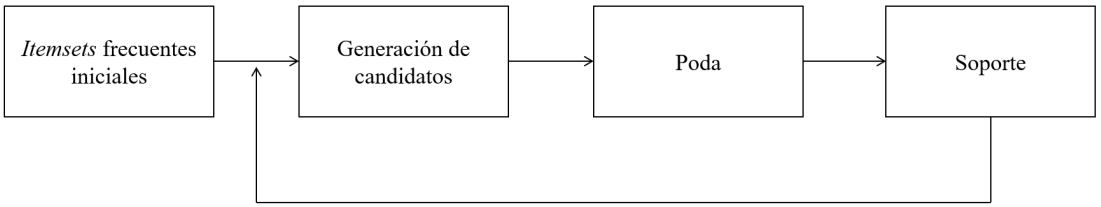
En este proyecto se implementará el algoritmo *a priori*. Este algoritmo encuentra reglas de asociación booleanas y de dimensión simple a partir de *itemsets* frecuentes.

El funcionamiento del algoritmo *a priori* se resume en dos pasos: en primer lugar, identifica conjuntos de *ítems* frecuentes y, en segundo lugar, genera reglas de asociación en base al nivel mínimo de confianza establecido (Moya Amaris y Rodríguez Rodríguez, 2003).

En la **identificación de *itemsets* frecuentes**, el principio *a priori* es fundamental. Este principio establece que un *itemset* solo puede ser frecuente si todos sus subconjuntos también lo son (Castro Casillas, 2016).

La Figura 3 ilustra el proceso que sigue el algoritmo *a priori* para la identificación de *itemsets* frecuentes. Se trata de un proceso iterativo, en el que en cada iteración se generan candidatos formados por *k* elementos, comenzando en *k* = 1. Dichos candidatos se someten a un proceso de poda, a través del cual se eliminan aquellos que no cumplen con la condición del principio *a priori*, es decir, aquellos cuyos subconjuntos no son frecuentes. De los resultantes, se calcula su soporte y se consideran frecuentes aquellos que superan el soporte mínimo establecido. El proceso finaliza cuando no se encuentran más conjuntos candidatos a ser frecuentes (Díaz Llerena, 2013).

Figura 3. Búsqueda de *itemsets* frecuentes.



Fuente: Díaz Llerena, 2013

Una vez se han identificado los conjuntos frecuentes, el algoritmo genera **reglas de asociación** que cumplen con el criterio mínimo de confianza establecido. Para cada conjunto frecuente *l*, se obtienen todos los posibles subconjuntos *s*. Para cada subconjunto *s*, se genera una regla $s \rightarrow (l - s)$ siempre y cuando se alcance el nivel mínimo de confianza establecido. Es decir, cuando el soporte (*l*) dividido entre el soporte (*s*) sea mayor o igual que el nivel mínimo de confianza (Moya Amaris y Rodríguez Rodríguez, 2003).

A continuación, se muestra en detalle el funcionamiento del algoritmo *a priori* a través de un ejemplo. En la Tabla 3 se presenta una base de datos que contiene información sobre transacciones realizadas por clientes en un supermercado.

Tabla 3. Ejemplo de una base de datos de compras realizadas en un supermercado.

Transacción	Ítems
t_1	Leche, café
t_2	Café, galletas
t_3	Café, arroz
t_4	Leche, café, galletas
t_5	Leche, arroz
t_6	Café, arroz
t_7	Leche, arroz
t_8	Leche, café, arroz, plátanos

t_9	Leche, café, arroz
t_{10}	Café, plátanos
t_{11}	Leche, galletas
t_{12}	Leche, café, galletas

Fuente: elaboración propia

En primer lugar, se identifican los *itemsets* frecuentes, asumiendo un soporte mínimo del 16%:

1. En la primera iteración se crean *itemsets* que contienen un único elemento, se calcula su soporte y, si cumplen el mínimo establecido, se consideran frecuentes. La Tabla 4 muestra un ejemplo de este primer paso.

Tabla 4. Identificación de *itemsets* frecuentes compuestos por un elemento.

Itemset	**Soporte**	**Frecuente**
{leche}	8/12 = 66,67%	Sí
{café}	9/12 = 75%	Sí
{arroz}	6/12 = 50%	Sí
{galletas}	4/12 = 33,33%	Sí
{plátanos}	2/12 = 16,67%	Sí

Fuente: elaboración propia

2. En la segunda iteración se generan *itemsets* formados por dos elementos, se calcula su soporte y se analiza si superan el mínimo establecido. En este caso, no se elimina ningún conjunto en el proceso de poda ya que todos los candidatos están formados por subconjuntos frecuentes. En la Tabla 5 se presentan los resultados del ejemplo propuesto.

Tabla 5. Identificación de *itemsets* frecuentes compuestos por dos elementos.

Itemset	**Soporte**	**Frecuente**
{leche, café}	5/12 = 41,67%	Sí
{leche, arroz}	4/12 = 33,33%	Sí
{leche, galletas}	3/12 = 25%	Sí
{leche, plátanos}	1/12 = 8,33%	No
{café, arroz}	4/12 = 33,33%	Sí
{café, galletas}	3/12 = 25%	Sí
{café, plátanos}	2/12 = 16,67%	Sí
{arroz, galletas}	0/12 = 0%	No
{arroz, plátanos}	1/12 = 8,33%	No
{galletas, plátanos}	0/12 = 0%	No

Fuente: elaboración propia

3. En la tercera iteración se generan candidatos formados por tres elementos. Sin embargo, como se puede observar en la Tabla 6, no se tienen en cuenta aquellos candidatos que contienen subconjuntos que no son frecuentes. Por ejemplo, el conjunto *{leche, café, plátanos}* se considera infrecuente ya que el *itemset {leche, plátanos}* no supera el soporte mínimo.

Tabla 6. Identificación de *itemsets* frecuentes compuestos por tres elementos.

Itemset	**Soporte**	**Frecuente**
{leche, café, arroz}	2/12 = 16,67%	Sí
{leche, café, galletas}	2/12 = 16,67%	Sí

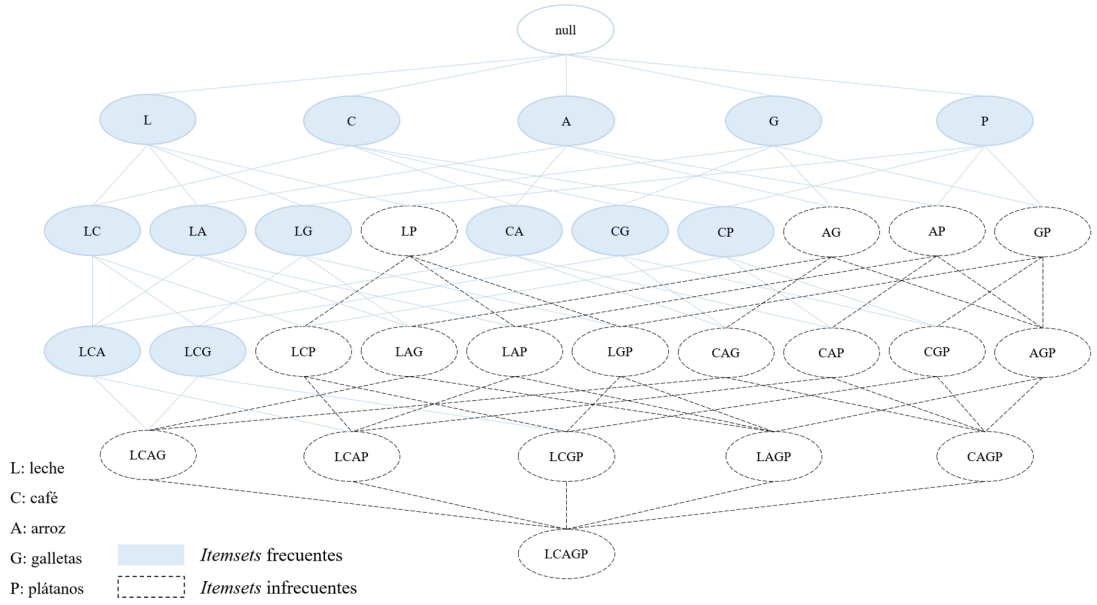
Fuente: elaboración propia

4. La búsqueda concluye al no encontrar candidatos de *itemsets* con cuatro elementos formados por subconjuntos frecuentes. Aunque podría considerarse el *itemset {leche, café, arroz, galletas}*, éste se descarta ya que está formado por subconjuntos infrecuentes, como por ejemplo *{arroz, galletas}*.

A continuación, la Figura 4 muestra un esquema en forma de árbol que ejemplifica el proceso de búsqueda de *itemsets* frecuentes y, por ende, el principio *a priori*. Los conjuntos frecuentes vienen representados por el color azul, mientras que las líneas discontinuas denotan los conjuntos que no son frecuentes.

Como se puede observar, siempre se cumple la condición del principio *a priori*, pues todos los conjuntos consecuentes formados por conjuntos infrecuentes, se consideran también infrecuentes.

Figura 4. Búsqueda de *itemsets* frecuentes.



Fuente: elaboración propia

Ahora bien, siguiendo el segundo paso del algoritmo, se procede a analizar, a modo de ejemplo, las posibles reglas de asociación del *itemset* frecuente *{leche, café, arroz}*. Considerando un nivel mínimo de confianza del 50%, la Tabla 7 indica que solamente son adecuadas las reglas *(leche, arroz → café)* y *(café, arroz → leche)*, pues son las únicas que alcanzan dicho nivel de confianza.

Por un lado, la primera regla indica que existe una probabilidad del 50% de que se compre café habiendo comprado leche y arroz conjuntamente. Por otro lado, la segunda regla establece que existe una probabilidad del 50% de que se compre leche cuando se compra café y arroz de manera conjunta.

Tabla 7. Análisis de posibles reglas de asociación del *itemset* frecuente {leche, café, azúcar}.

Regla de asociación	Confianza	Adecuada
leche → café, arroz	0,1667/0,6667 = 25%	No
café → leche, arroz	0,1667/0,75 = 22,23%	No
arroz → leche, café	0,1667/0,5 = 33,34%	No
leche, café → arroz	0,1667/0,4167 = 40%	No
leche, arroz → café	0,1667/0,3333 = 50%	Sí
café, arroz → leche	0,1667/0,3333 = 50%	Sí

Fuente: elaboración propia

Así pues, el algoritmo *a priori* es capaz de generar reglas de asociación a partir de conjuntos de *ítems* frecuentes. Sin embargo, existe una gran limitación. A medida que el tamaño de la base de datos aumenta, el rendimiento del algoritmo disminuye. Esto se debe a que, en cada iteración, el algoritmo recorre la base de datos completa con el fin de actualizar los soportes y descartar aquellos conjuntos infrecuentes para futuras iteraciones (Díaz-Molina y García-Garrido, 2018).

El algoritmo *a priori* utiliza una técnica de búsqueda exhaustiva para identificar grandes conjuntos de *ítems* candidatos a ser frecuentes. Sin embargo, esta metodología se enfrenta a desafíos computacionales cuando la capacidad de memoria es limitada y la base de datos es muy extensa (Narvekar y Syed, 2015).

Una alternativa que consigue superar esta desventaja es el algoritmo *Frequent Pattern* (FP) *growth*. Este algoritmo emplea una estructura de árbol de patrones frecuentes que almacena información sobre *itemsets* frecuentes de forma compacta. Esto reduce significativamente el coste computacional ya que no es necesario construir conjuntos de *ítems* candidatos a ser frecuentes, ni evaluar si superan un umbral específico.

A diferencia del algoritmo *a priori*, el árbol del algoritmo FP no está formado por un número exponencial de nodos. El árbol comienza con un nodo raíz y genera un conjunto de subárboles que almacenan los valores de las transacciones. Además, utiliza una tabla que lleva un registro sobre los elementos que se acceden con mayor frecuencia (Orozco Bohórquez, 2017).

2.4. Instacart.

La base de datos utilizada para este proyecto proviene de la compañía Instacart. Instacart es una empresa que lidera el sector de compra *online* de alimentos y productos en tiendas locales y supermercados en Norteamérica. La compañía colabora con más de 500 proveedores, incluyendo Aldi, Costco, Food Lion y Loblaws. Entre Estados Unidos y Canadá, Instacart abarca cerca de 40.000 tiendas físicas.

Si bien la empresa se fundó en 2012, no fue hasta la pandemia del COVID-19 cuando creció de forma exponencial. Durante este periodo, su volumen de pedidos aumentó en un 500%, en comparación con los volúmenes alcanzados en las mismas fechas del año anterior. Este fenómeno surge como consecuencia del enorme incremento en el número de compradores, pasando de 200.000 en marzo a 500.000 a finales de año.

En su modelo de negocio, Instacart distingue dos tipos de usuarios: el consumidor y el comprador. Por un lado, el consumidor realiza su pedido *online* a través de la página web o aplicación móvil de Instacart. Indica su código postal y selecciona la tienda disponible en la que desea comprar. Una vez elegidos los productos, el consumidor establece su preferencia de entrega, ya sea en dos horas, una semana, o en una fecha futura.

Por otro lado, el comprador es quien acude a las tiendas, realiza la compra y entrega el pedido al consumidor. Puede decidir trabajar en cualquier momento iniciando sesión en la aplicación y seleccionando los pedidos que va a realizar. Sin embargo, en algunas tiendas donde el volumen de pedidos es muy alto, existen dependientes contratados a tiempo parcial para garantizar la ejecución de todos los pedidos.

Durante el proceso de compra, tanto el comprador como el consumidor mantienen el contacto a través de un *chat*. Esto permite al consumidor comunicar sus preferencias en cuanto a productos y seguir el proceso de compra a medida que el comprador escanea los artículos que añade a la cesta.

Un elemento crucial que asegura el funcionamiento eficiente de Instacart son sus modelos de *Machine Learning*. Dado que la empresa no gestiona tiendas ni almacenes, el control de inventario supone un reto. Por ello, han desarrollado un modelo que

predice las probabilidades de que un artículo esté disponible en un momento determinado.

Si un consumidor agrega a su cesta un producto cuyas probabilidades de estar disponible son muy bajas, un modelo de recomendación sugiere al consumidor una lista de artículos sustitutivos. El usuario elegirá una de las alternativas en caso de que el producto principal no se encuentre en la tienda.

La aplicación también cuenta con un algoritmo que equilibra a tiempo real el número de compradores disponibles con la demanda de pedidos. También, asigna los pedidos al comprador que considera más adecuado en función de características como la edad, entre otras. Por ejemplo, si el pedido contiene artículos como alcohol, el comprador no podrá ser menor de edad.

Además, se ha implementado otro modelo que calcula la capacidad de entrega de Instacart a lo largo del día. Es decir, estima el número de compradores disponibles para realizar la compra y entrega de los pedidos.

Con el objetivo de optimizar el proceso de entrega, Instacart ha desarrollado dos modelos adicionales. Uno de ellos estima la hora de llegada del comprador a la tienda y otro el tiempo de estacionamiento hasta que se dirige al destino.

Una vez que el comprador está preparado para realizar la entrega, entra en juego un algoritmo que diseña la ruta más eficiente utilizando información del tráfico a tiempo real. Gracias a este algoritmo, el consumidor puede saber la hora de entrega estimada.

Así pues, Instacart se apoya en algoritmos inteligentes y tecnologías que le permiten ofrecer un servicio eficiente a sus usuarios en el proceso de compra y entrega de alimentos y productos de tiendas locales y supermercados (Rao y Zhang, 2021).

3. Estudio de campo.

En el estudio de campo se profundizará en el análisis de las transacciones de la base de datos de Instacart, así como en descubrir patrones de compra mediante la identificación de conjuntos frecuentes con un nivel de confianza mínimo.

Este análisis se traducirá en la generación de *insights* significativos sobre el comportamiento de los consumidores, lo cual resultará fundamental para el desarrollo de estrategias comerciales efectivas en la plataforma Instacart.

3.1. *Dataframe*: muestra y variables seleccionadas.

El *dataframe* utilizado contiene información acerca de los pedidos realizados por más de 200.000 clientes a través de la plataforma Instacart. Para aplicar el algoritmo *a priori*, sólo es necesario conocer el identificador de cada pedido y la lista de artículos comprados en dicho pedido. Sin embargo, con el objetivo de llevar a cabo un análisis más completo, se creará un *dataframe* mediante la combinación de diversas bases de datos, que permitirá obtener información adicional sobre los pedidos.

Los conjuntos de datos utilizados provienen de la página web *Kaggle* (Psparks, 2017) y están completamente limpios. En la Tabla 8 se muestra información sobre los *datasets* encontrados.

Tabla 8. *Datasets*.

Dataset	**Información**	**Variables**
Products	Productos que se venden a través de Instacart.	*product_id*, *product_name*, *aisle_id*, *department_id*
Aisles	Pasillo donde se colocan los productos.	*aisle_id*, *aisle*
Departments	Departamentos a los que pertenecen los productos.	*department_id*, *department*
Order_products_prior	Productos comprados en pedidos históricos de los clientes. En este caso, se identifican entre tres y 100 pedidos por cliente.	*order_id*, *product_id*, *add_to_cart_order*, *reordered*

Order_products_train	Productos comprados en pedidos históricos de los clientes. En este caso, se identifica 1 pedido por cliente.	*order_id*, *product_id*, *add_to_cart_order*, *reordered*
Orders	Información detallada de los pedidos realizados.	*order_id*, *user_id*, *eval_set*, *order_number*, *order_dow*, *order_hour_of_day*, *days_since_prior_order*

Fuente: elaboración propia

A partir de la combinación de algunas de estas bases de datos, se crearán dos *dataframes*. El primero, para el análisis exploratorio de los datos y, el segundo, para la aplicación del algoritmo *a priori*.

Como se puede observar, "*orders_products_prior*" y "*orders_products_train*" tienen las mismas variables, pero no el mismo número de observaciones. Por un lado, en "*order_products_prior*" se contempla más de un pedido por cliente y, por tanto, el número total de observaciones supera los 32 millones. Por otro lado, "*orders_products_train*" identifica un pedido por cliente y, por tanto, contempla aproximadamente un millón y medio de observaciones.

Así pues, para el algoritmo *a priori* utilizaremos esta última base de datos ya que, como bien se ha explicado anteriormente, el rendimiento del algoritmo disminuye a medida que el tamaño del *dataset* aumenta. Es por ello por lo que se prefiere una base de datos de menor dimensión.

Sin embargo, para el análisis y exploración de los datos se utilizará un *dataset* de mayor tamaño. Esto permitirá obtener más información sobre los datos de la manera más completa y precisa posible.

Los dos *dataframes* se obtendrán mediante la combinación de los *datasets* explicados previamente. Así pues, con el objetivo de entender la lógica de las combinaciones, será necesario comprender el contenido de las variables de los diferentes *datasets*.

En la Tabla 9 se detallan las variables del conjunto de datos "*products*". A partir de este *dataset* se puede inferir que Instacart cuenta con una amplia variedad de productos, concretamente con 49.688 artículos. Cada uno de estos productos se encuentra en uno de los 134 pasillos y pertenece a uno de los 21 departamentos disponibles.

Tabla 9. Variables del *dataset* "products".

Variable	Concepto	Tipo	Rango de valores
product_id	Identificador de cada producto.	Número entero	1-49.688
product_name	Nombre de cada producto.	Factor	"Chocolate Sandwich Cookies", "Dry Nose Oil", …
aisle_id	Identificador del pasillo al que pertenece cada producto.	Número entero	1-134
department_id	Identificador del departamento al que pertenece cada producto.	Número entero	1-21

Fuente: elaboración propia

En la Tabla 10 se explican las variables de la base de datos "*aisles*". Este conjunto de datos identifica 134 pasillos y sus respectivos nombres.

Tabla 10. Variables del *dataset* "aisles".

Variable	Concepto	Tipo	Rango de valores
aisle_id	Identificador de cada pasillo.	Número entero	1-134
aisle	Nombre de cada pasillo.	Factor	"instant foods", "energy granola bars", …

Fuente: elaboración propia

En la Tabla 11 se exponen las variables de la base de datos "*departments*". Este *dataset* identifica 21 departamentos y sus respectivos nombres.

Tabla 11. Variables del *dataset* "departments".

Variable	Concepto	Tipo	Rango de valores
department_id	Identificador de cada departamento.	Número entero	1-21
department	Nombre de cada departamento.	Factor	"frozen", "beverages", …

Fuente: elaboración propia

En la Tabla 12 se detallan las variables de la base de datos "*order__product_prior*". Este conjunto de datos asocia cada producto comprado con su respectivo pedido. A partir de la variable "*add_to_cart_order*" se puede determinar que el número máximo de artículos encontrados en un pedido es 145.

Tabla 12. Variables del *dataset* "order_products_prior".

Variable	Concepto	Tipo	Rango de valores
order_id	Identificador de cada pedido.	Número entero	2-3.421.083
product_id	Identificador de los productos que aparecen en cada pedido.	Número entero	1-49.688
add_to_cart_ order	Orden en el que se han añadido los productos al pedido.	Número entero	1-145
reordered	Si el producto se ha vuelto a pedir.	Binario	0: no se ha recomprado 1: sí se ha recomprado

Fuente: elaboración propia

En la Tabla 13 se exponen las variables de la base de datos "*order_product_train*". Este conjunto de datos muestra la misma información que "*order_product_prior*", pero como se ha explicado anteriormente, el número de observaciones es mucho menor. Además, en este caso, la variable "*add_to_cart_order*" muestra que el número máximo de artículos encontrados en un pedido es 80.

Tabla 13. Variables del *dataset* "order_products_train".

Variable	Concepto	Tipo	Rango de valores
order_id	Identificador de cada pedido.	Número entero	1-3.421.070
product_id	Identificador de los productos que aparecen en cada pedido.	Número entero	1-49.688
add_to_cart_order	Orden en el que se han añadido los productos al pedido.	Número entero	1-80
reordered	Si el producto se ha vuelto a pedir.	Binario	0: no se ha recomprado 1: sí se ha recomprado

Fuente: elaboración propia

Finalmente, en la Tabla 14 se explican las variables de la base de datos "*orders*". Este *dataset* aporta información detallada de cada pedido. Se observa que los pedidos han sido realizados por 206.209 clientes, que el número máximo de pedidos realizados por un cliente ha sido 100, y que el número máximo de días entre dos pedidos ha sido 30.

Tabla 14. Variables del *dataset* "orders".

Variable	Concepto	Tipo	Rango de valores
order_id	Identificador de cada pedido.	Número entero	1-3.421.083
user_id	Identificador de cada cliente.	Número entero	1-206.209
eval_set	Conjunto de datos al que pertenecen los pedidos.	Factor	"prior", "train", "test"
order_number	Número de pedido para cada cliente.	Número entero	1-100
order_dow	Día de la semana en la que se ha realizado el pedido.	Factor	0: domingo; 1: lunes; 2: martes; 3: miércoles; 4: jueves; 5: viernes; 6: sábado

order_hour_of_day	Hora del día a la que se ha realizado el pedido.	Factor	"00", "01", "02", …
days_since_prior_order	Diferencia de días entre dos pedidos.	Número entero	1-30

Fuente: elaboración propia

Una vez entendidas todas las variables, se crea el primer *dataframe*, es decir, el que se utilizará para el análisis exploratorio. Este *dataframe* surge de las siguientes combinaciones:

1. Unión de "*products*" y "*asiles*" por la variable "*aisle_id*", para conocer el nombre del pasillo en el que se encuentra cada producto.
2. Unión de la base de datos resultante y "*departments*" por la variable "*department_id*", para conocer también el nombre del departamento al que pertenece cada producto.
3. Unión del *dataset* resultante y "*order_products_prior*" por la variable "*product_id*", para conocer, de cada pedido, el nombre, pasillo y departamento de los productos comprados.
4. Unión de la base de datos resultante y "*orders*" por la variable "*order_id*", para obtener información específica acerca de los pedidos, como por ejemplo el cliente que lo ha realizado o el día y la hora en la que ha sido realizado.

Por lo tanto, este primer *dataframe* contiene 32.434.489 observaciones y 15 variables. Las variables que lo componen son las siguientes: "*order_id*", "*product_id*", "*department_id*", "*aisle_id*", "*product_name*", "*aisle*", "*department*", "*add_to_cart_order*", "*reordered*", "*user_id*", "*eval_set*", "*order_number*", "*order_dow*", "*order_hour_of_day*", y "*days_since_prior_order*".

Es importante destacar que la variable "*eval_set*" solo tomará el valor "prior" ya que la base de datos se ha construido con el conjunto "*order_products_prior*".

Para el segundo *dataframe*, es decir, el que se utilizará para la implementación del algoritmo *a priori*, solo es necesario conocer el identificador de los pedidos y el nombre de los productos. La creación del *dataframe* implica los siguientes pasos:

1. Unión de "*products*" y "*order_products_train*" por la variable "*product_id*" con el objetivo de obtener información sobre el nombre de los productos asociados a los pedidos.
2. Se seleccionan únicamente las variables "*order_id*" y "*product_name*" ya que son las únicas necesarias para el algoritmo.

Así pues, este segundo *dataframe* se compone de 1.384.617 observaciones y dos variables, "*order_id*" y "*product_name*".

3.2. Análisis exploratorio de los datos.

A continuación, se llevará a cabo un análisis exhaustivo del primer *dataframe* con el fin de profundizar en la comprensión de los datos y obtener conclusiones más sólidas una vez el algoritmo *a priori* encuentre reglas de asociación.

En primer lugar, la Figura 5 revela que la mayoría de los clientes han efectuado entre 4 y 10 pedidos. Además, se puede apreciar que aproximadamente 2.500 clientes han realizado hasta 100 pedidos en Instacart.

Por otro lado, en la Figura 6 se observa que los días con mayor número de pedidos son los domingos y los lunes, a diferencia de los jueves, que muestran el menor volumen de pedidos realizados.

La Figura 7 evidencia que la mayoría de los pedidos se efectúan entre las diez de la mañana y las cuatro de la tarde. Sin embargo, a partir de las cinco de la tarde se observa una notable disminución en el número de pedidos.

En la Figura 8, al combinar el día de la semana y las horas, se observa que la mayoría de los productos fueron adquiridos los domingos, principalmente entre las diez de la mañana y las cuatro de la tarde, y los lunes, especialmente de nueve a once de la mañana. También, cabe destacar que en el resto de los días de la semana, los artículos fueron mayoritariamente comprados durante el día, es decir, entre las ocho de la mañana y las cinco de la tarde. El gráfico presenta de manera clara las franjas horarias y los días de la semana en los que se han realizado la mayoría de las compras.

La Figura 9 destaca que la mayoría de los clientes realizan un nuevo pedido una vez transcurridos siete o treinta días desde su última compra. Además, se observa una

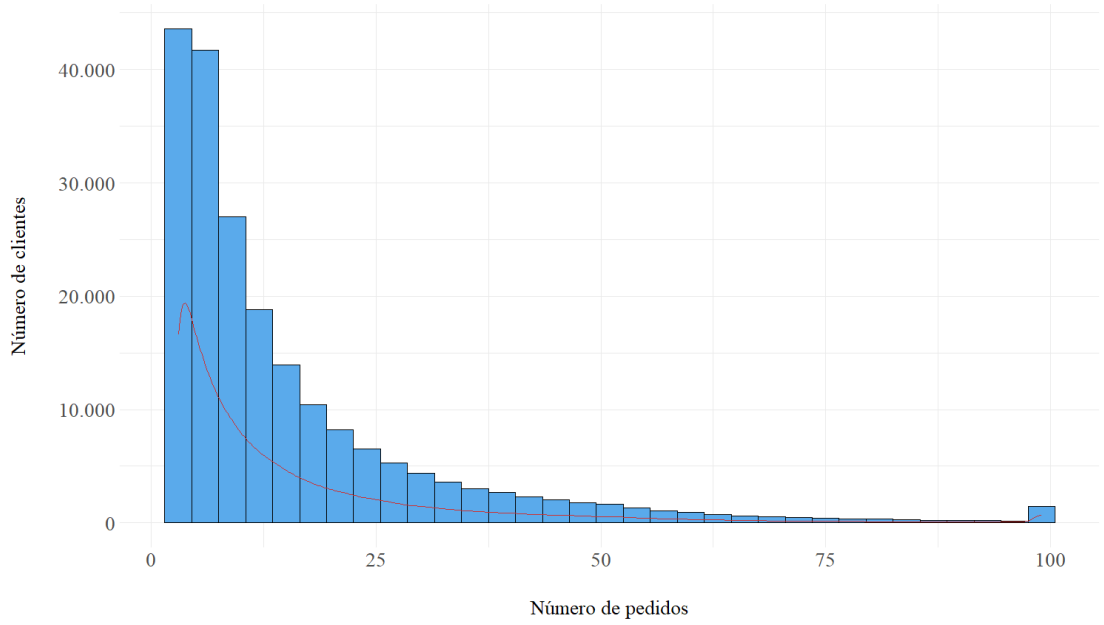
notable cantidad de pedidos que se efectúan durante la primera semana inmediatamente después de realizar un pedido.

En la Figura 10, se presentan los 12 productos más comprados y los 12 más recomprados. Es evidente que los productos más comprados coinciden con aquellos que son más frecuentemente recomprados. Estos son los plátanos, plátanos orgánicos, fresas, espinacas y aguacate.

La Figura 11 expone la distribución de los pedidos según el número de artículos que contienen. Se puede observar claramente que la mayoría de los pedidos se concentran en el extremo inferior, es decir, su tamaño es reducido. La mayor parte de los pedidos incluyen entre cinco y diez productos. Además, cabe destacar que la cola de la distribución es muy extensa debido a la presencia de un pedido que consta de 145 artículos.

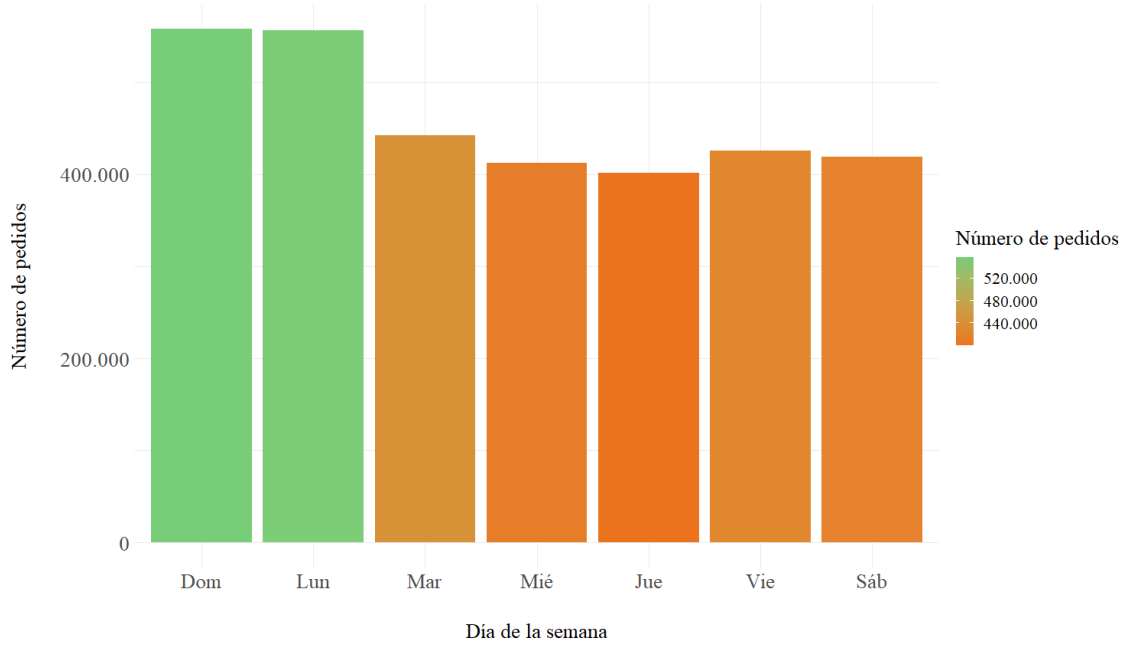
Finalmente, en la Figura 12 se presentan los diez pasillos más visitados y los diez departamentos más demandados. Por un lado, los pasillos que los clientes frecuentan en mayor medida se centran en frutas y verduras. Por otro lado, los departamentos más populares incluyen aquellos que se dedican a productos frescos, productos lácteos y huevos, y bebidas.

Figura 5. Distribución de pedidos por cliente.



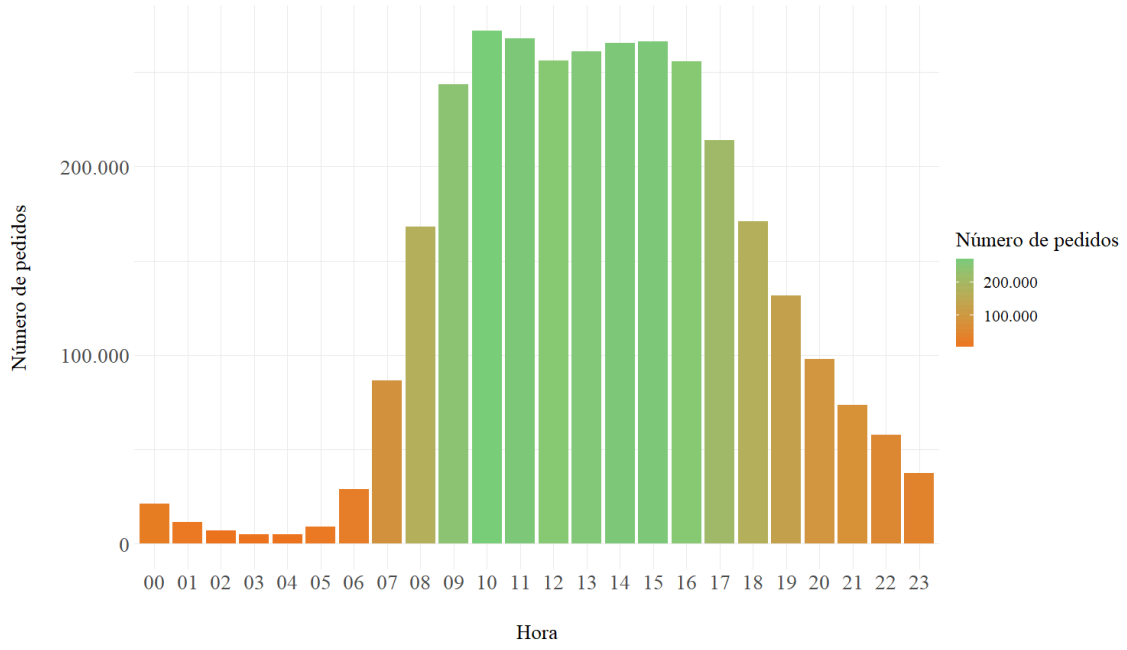
Fuente: elaboración propia con R Studio

Figura 6. Número de pedidos por día de la semana.



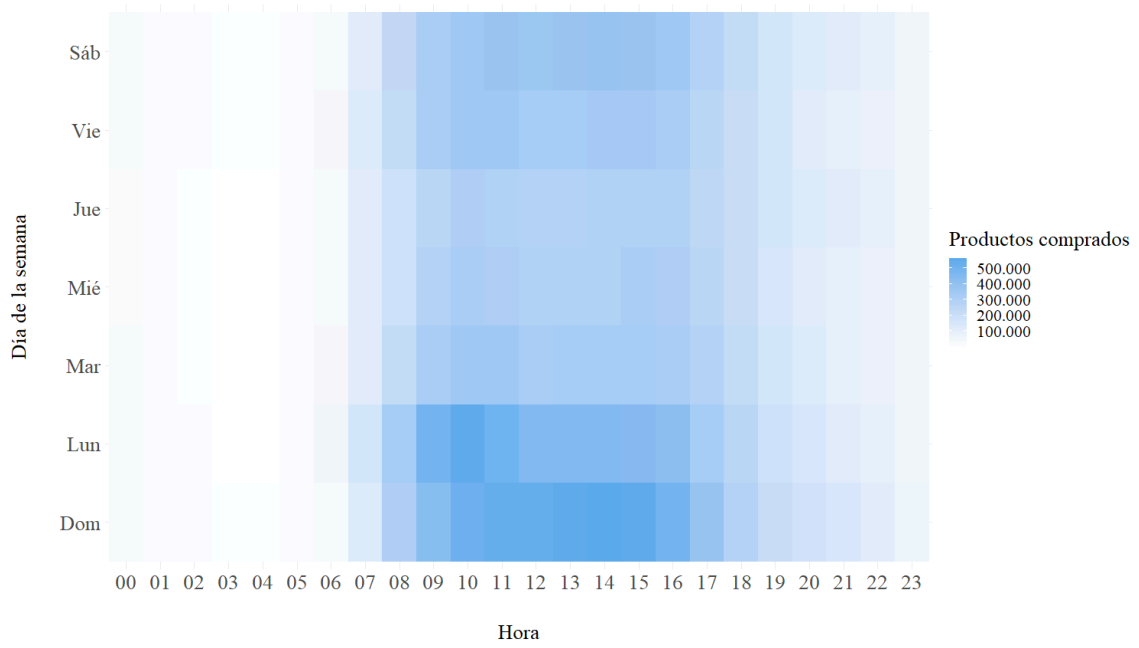
Fuente: elaboración propia con R Studio

Figura 7. Número de pedidos por hora del día.



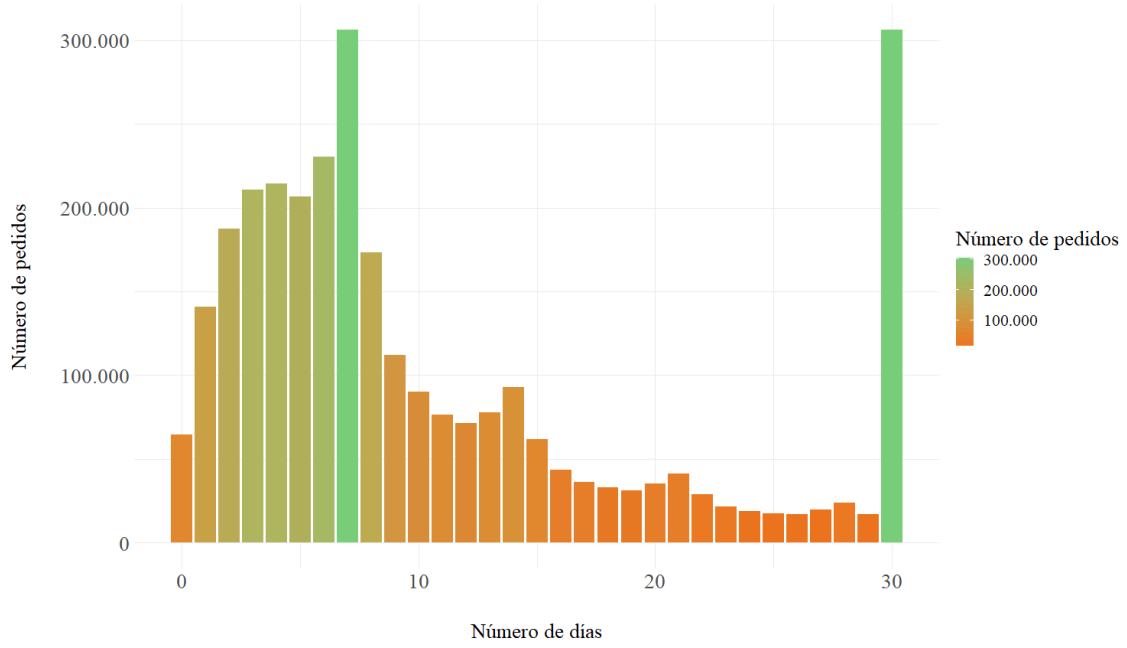
Fuente: elaboración propia con R Studio

Figura 8. Número de productos comprados por día y hora.



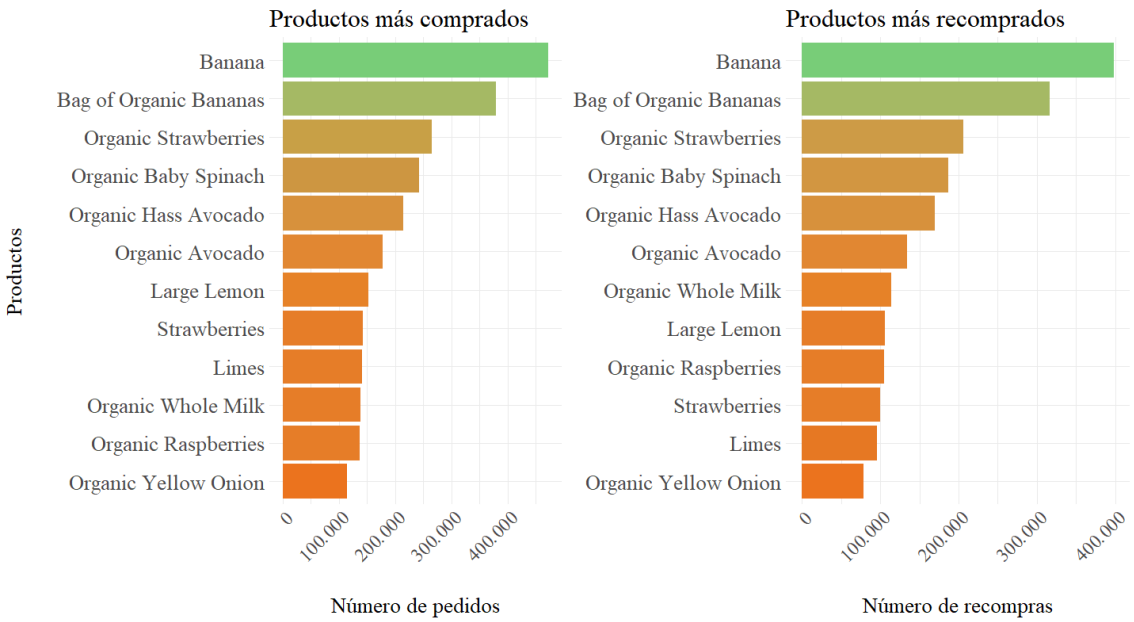
Fuente: elaboración propia con R Studio

Figura 9. Distribución de pedidos por el número de días transcurridos desde el último pedido.



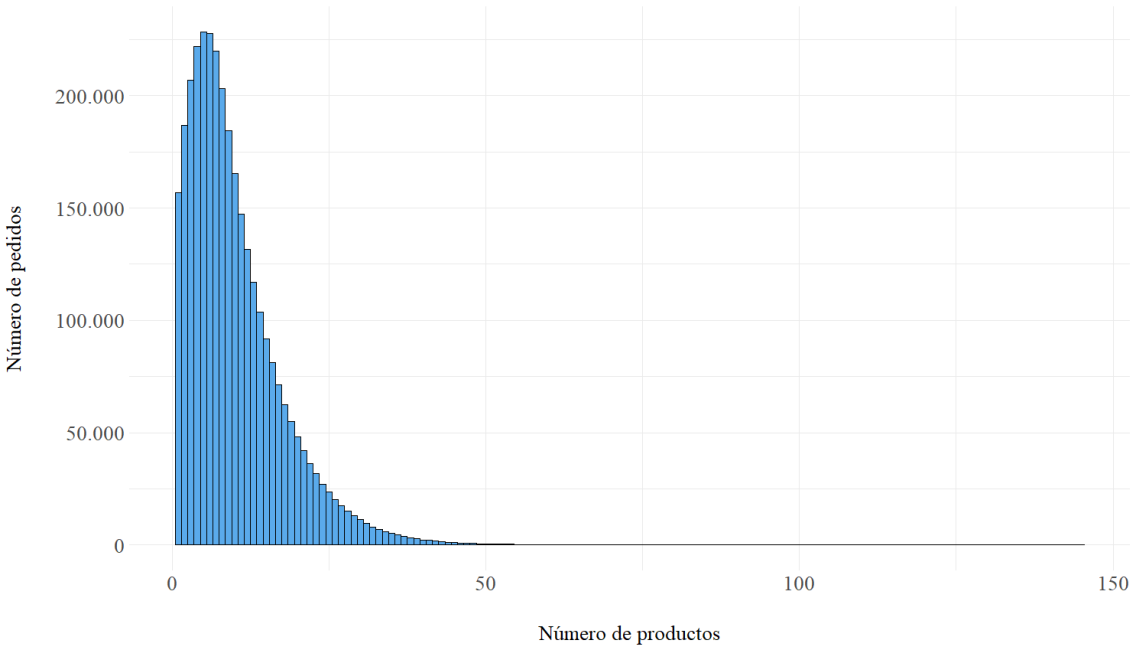
Fuente:elaboración propia con R Studio

Figura 10. Artículos populares.



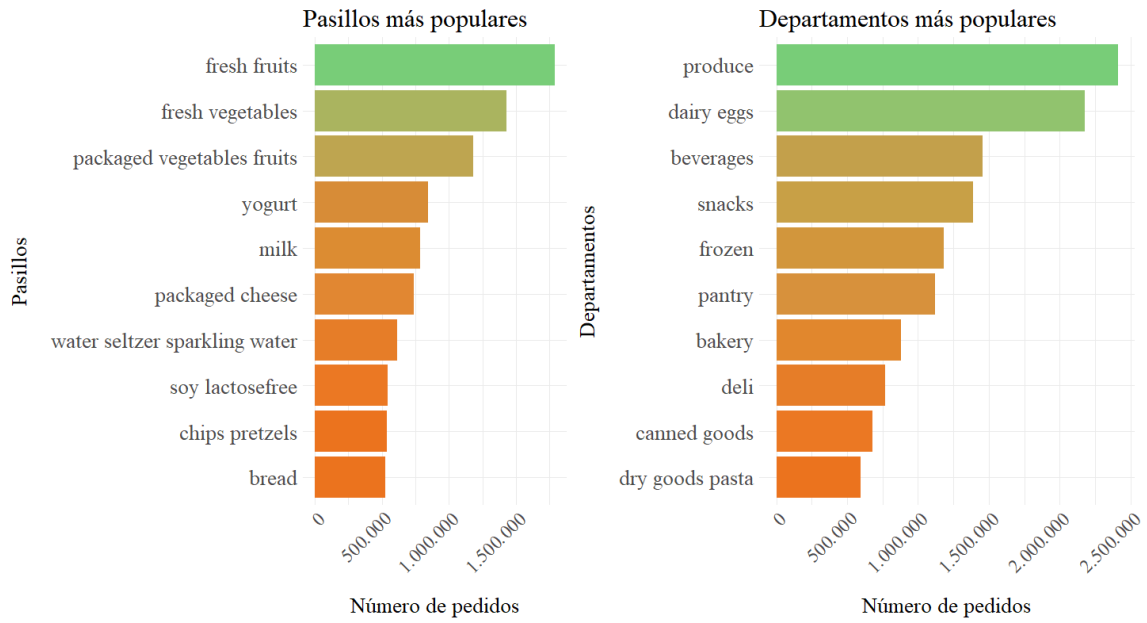
Fuente: elaboración propia con R Studio

Figura 11. Distribución de pedidos en función de su tamaño.



Fuente: elaboración propia con R Studio

Figura 12. Pasillos y departamentos más populares.



Fuente: elaboración propia con R Studio

3.3. Aplicación del algoritmo *a priori*.

Una vez realizado el análisis exploratorio del primer *dataframe*, se implementará el algoritmo *a priori* utilizando el segundo *dataframe*. Las dimensiones de este último *dataframe* son más reducidas debido a la limitación computacional del algoritmo, que se acentúa con bases de datos más extensas.

Así pues, el primer paso a realizar consiste en **transformar la base de datos a la clase "transacciones"** con el fin de identificar cada pedido y sus respectivos productos. En el Anexo I se muestra el código en R utilizado para la transformación del dato.

El conjunto de datos resultante identifica un total de 131.209 transacciones y 39.123 productos. El coste de oportunidad al utilizar un *dataframe* de menor tamaño implica perder parte de información. En este caso, se consideran 39.123 productos de un total de 49.688. Esto significa que tenemos en cuenta aproximadamente el 80% del total de los productos.

A continuación, la Tabla 15 presenta un resumen del tamaño de las transacciones. En promedio, se observa que el número de artículos por pedido oscila entre 10 y 11. El

número máximo de artículos en un pedido es 80, y el mínimo uno. Existe un 50% de transacciones que incluyen entre cinco y 14 artículos.

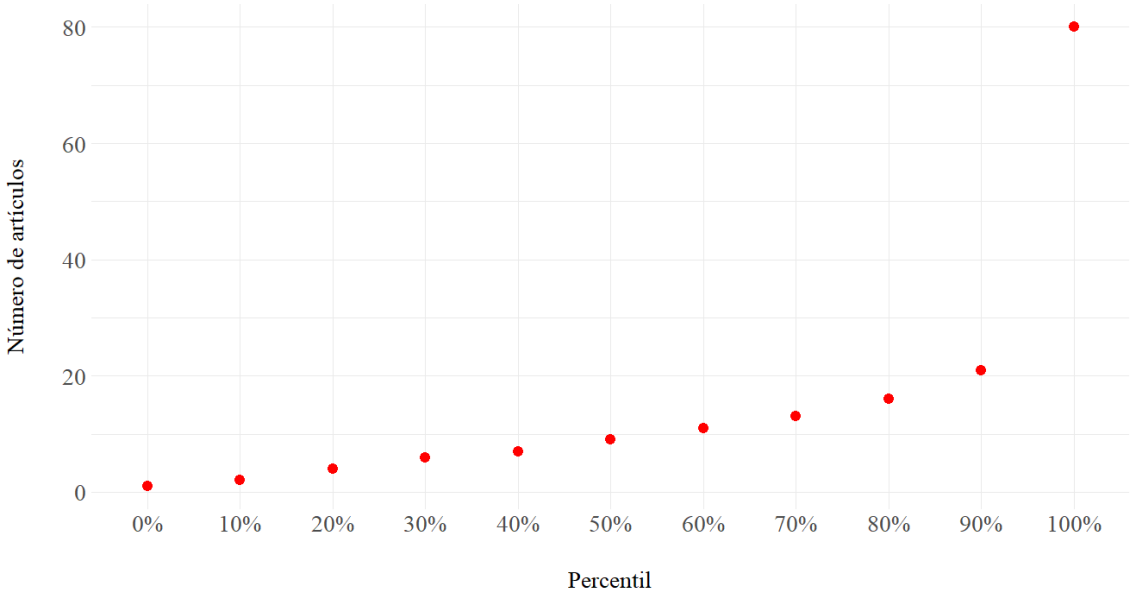
Tabla 15. Tamaño de las transacciones.

Mínimo	1º Cuartil	Mediana	Media	3º Cuartil	Máximo
1	5	9	10,55	14	80

Fuente: elaboración propia con R Studio

La Figura 13 muestra los percentiles del tamaño de las transacciones en intervalos del 10%. Así pues, se observa que el 90% de las transacciones contienen como máximo 21 productos y que existe un 10% que contienen entre 21 y 80 artículos.

Figura 13. Percentiles del tamaño de las transacciones.



Fuente: elaboración propia con R Studio

El segundo paso consiste en identificar los ***itemsets* frecuentes** dentro del conjunto de transacciones. Para determinar si un *itemset* es frecuente, se debe establecer un soporte mínimo. En este caso, el soporte mínimo se define como la media de los soportes de los *itemsets* de tamaño i =1, es decir, la frecuencia relativa media de los conjuntos de *ítems*

formados por un único elemento. Por ende, el soporte mínimo establecido se sitúa en el 0,027%.

Una vez determinado el soporte mínimo, se obtienen los *itemsets* frecuentes. En el Anexo I se muestra el código utilizado para la identificación de estos conjuntos frecuentes. Nótese que se utiliza la función "*apriori*" de la librería "*arules*". Esta función permite modificar los valores establecidos por defecto de parámetros como el soporte mínimo, la confianza mínima, el número máximo de elementos en un conjunto, y el tiempo máximo permitido para verificar los subconjuntos creados.

Esta función se emplea para obtener tanto los conjuntos de elementos frecuentes como las reglas de asociación. El parámetro "*target*" se utiliza para especificar aquello que se desea identificar. En este caso, se busca identificar los conjuntos frecuentes, por lo que se indica *target = "frequent itemset"*.

El tercer paso consiste en descubrir **reglas de asociación** a partir de los *itemsets* frecuentes. Como se mencionaba anteriormente, para que se formalice una regla de asociación debe cumplirse lo siguiente:

1. El conjunto de *ítems* debe ser considerado frecuente, lo que implica alcanzar el soporte mínimo que sitúa en un 0,027%.
2. El conjunto de *ítems* debe satisfacer el nivel de confianza mínimo. En este caso, se ha establecido en un 70%.

Ahora bien, una vez definidos el soporte mínimo y la confianza mínima, se identifican las reglas de asociación. Para ello, como se muestra en el Anexo I, se emplea la función "*apriori*" de nuevo, aunque esta vez indicando el parámetro *target = "rules"*.

3.4. Limitaciones encontradas.

Como bien se ha mencionado en apartados anteriores, el algoritmo *a priori* presenta una limitación computacional ya que debe recorrer toda la base de datos cada vez que busca conjuntos frecuentes. Esto puede suponer una gran carga a nivel de recursos y tiempo debido a las exigencias informáticas que supone implementar el algoritmo con bases de datos extensas.

De este modo, la razón principal por la que se utilizan *dataframes* diferentes en el análisis exploratorio y en la aplicación del algoritmo ha sido precisamente dicha limitación. Al intentar aplicar el algoritmo con el primer *dataframe*, se experimentó una carga computacional considerable, lo que llevó a optar por otras alternativas más eficientes como el uso del segundo *dataframe*.

Esta decisión ha traído consigo ciertos inconvenientes. Entre ellos, la pérdida de información. Como se mencionaba anteriormente, a la hora de buscar reglas de asociación sólo se tiene en cuenta aproximadamente el 80% del conjunto total de artículos disponibles. Esto significa que se pierde la oportunidad de descubrir reglas que podrían asociar productos que forman parte del 20% restante.

Esta casuística podría suponer implicaciones negativas en la precisión de las conclusiones. Al perder información de una quinta parte de los productos, se descartan potenciales patrones o conexiones relevantes entre los mismos.

Por otro lado, la función "*apriori*" utilizada en R para la aplicación del algoritmo contempla el parámetro "*maxtime*". Este parámetro permite determinar el tiempo máximo en segundos que se emplea para verificar los subconjuntos que han sido creados. Cuanto mayor sea el valor de este parámetro, mayor será la carga computacional ya que el algoritmo dedicará más tiempo a la verificación de los subconjuntos encontrados.

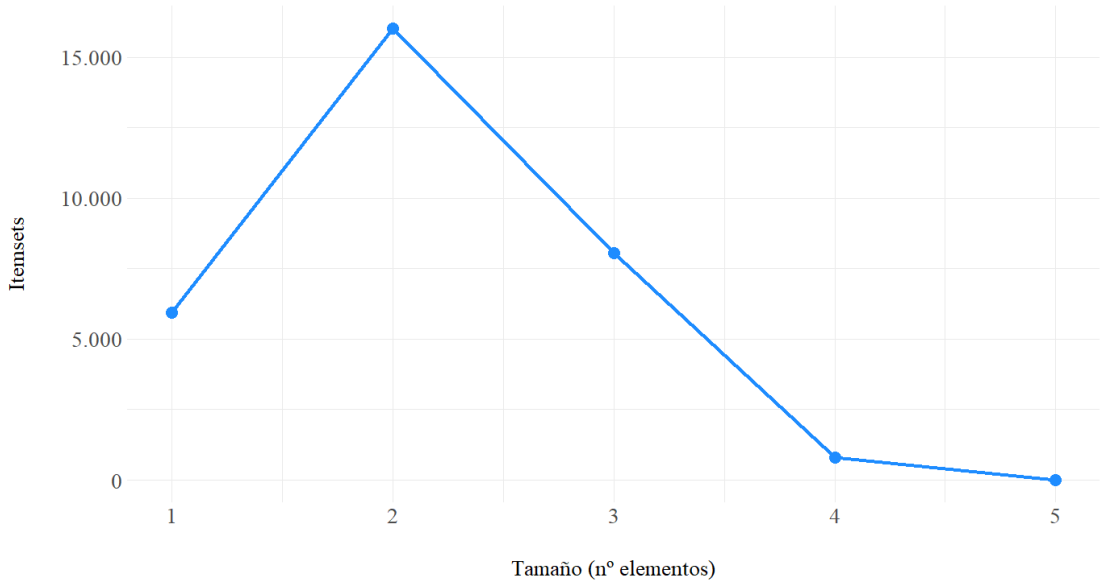
4. Resultados.

A continuación, se exponen los resultados obtenidos a partir de la implementación del algoritmo *a priori* sobre la base de datos de Instacart. Por un lado, se comentarán los conjuntos frecuentes encontrados y, por otro lado, las reglas de asociación identificadas.

Como bien se mencionaba anteriormente, estos resultados permitirán a Instacart tomar decisiones de carácter estratégico con el propósito de mejorar su posición en el mercado.

En primer lugar, se han encontrado **30.818 conjuntos frecuentes**, lo que significa que 30.818 *itemsets* superan el nivel de soporte mínimo (0,027%). La Figura 14 muestra la distribución de los mismos en función de su tamaño, es decir, del número de elementos que contienen los *itemsets*. Como se puede observar, más de la mitad, concretamente 16.030, están formados por dos elementos, mientras que solamente ocho contienen cinco elementos.

Figura 14. Tamaño de los *itemsets* identificados.



Fuente: elaboración propia con R Studio

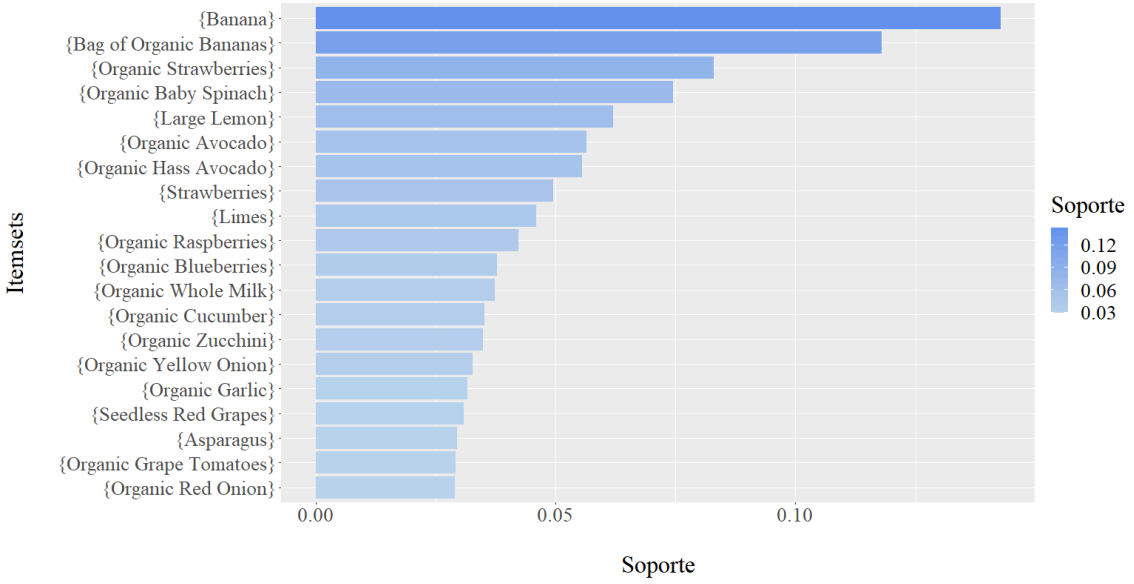
Ahora bien, la Figura 15 muestra los 20 *itemsets* más frecuentes identificados por el algoritmo *a priori*, es decir, aquellos que tienen el mayor soporte. Como era de esperar, estos *itemsets* coinciden con los artículos más populares mostrados en la Figura 10 del

apartado 3, pues en ambos casos se estudia la frecuencia de los mismos. Sin embargo, en este caso se analiza la frecuencia relativa, es decir, la frecuencia con la que aparecen los artículos con respecto al número total de transacciones.

Como se puede observar en la Figura 15, los cinco *itemsets* que presentan un mayor soporte son {plátano}, {bolsa de plátanos ecológicos}, {fresas ecológicas}, {espinacas pequeñas ecológicas} y {limón grande}, con un soporte de 14,27%, 11,80%, 8,30%, 7,46% y 6,20%, respectivamente. Esto significa que, por ejemplo, el plátano aparece en un 14,27% de las compras realizadas a través de Instacart.

También, cabe destacar que los 20 *itemsets* más frecuentes están compuestos por un único elemento. Esto resulta lógico ya que existe una probabilidad menor de que dos o más artículos aparezcan de forma conjunta en más transacciones que un solo artículo, aunque cabe destacar que tal situación podría haber ocurrido.

Figura 15. Los 20 *itemsets* más frecuentes.



Fuente: elaboración propia con R Studio

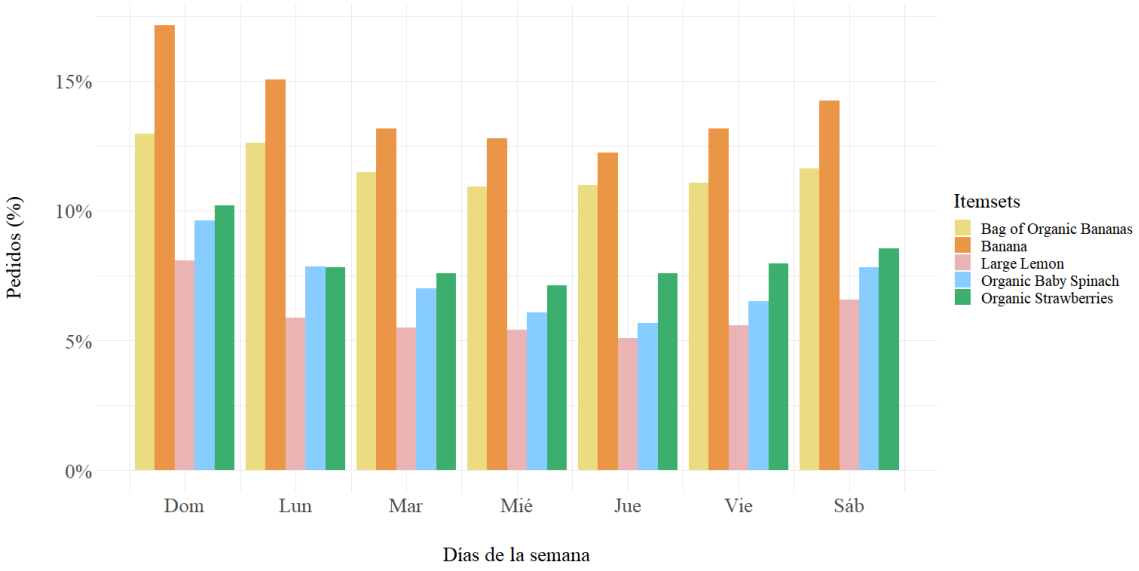
A continuación, la Figura 16 muestra, para cada día de la semana, el porcentaje de pedidos que contienen cada uno de los cinco *itemsets* más frecuentes. Por ejemplo, de los pedidos realizados el miércoles, el 5,40% contienen el *itemset* {limón grande}.

Como se puede observar, en todos los casos, el día que más pedidos se realiza es el domingo, como ya mostraba la Figura 6 del apartado 3, y el *itemset* que más se compra es {plátano}.

Por otro lado, el *itemset* {fresas ecológicas} se compra siempre más que el *itemset* {espinacas pequeñas ecológicas}, excepto el lunes, que se realizan más pedidos que contienen {espinacas pequeñas ecológicas}.

Además, todos los *itemsets* siguen el mismo patrón de compra a lo largo de la semana. Como se puede observar, los fines de semana se registran la mayor cantidad de pedidos, mientras que entre semana la cantidad de pedidos realizados es menor.

Figura 16. Número de pedidos por día de la semana de los cinco *itemsets* más frecuentes.



Fuente: elaboración propia con R Studio

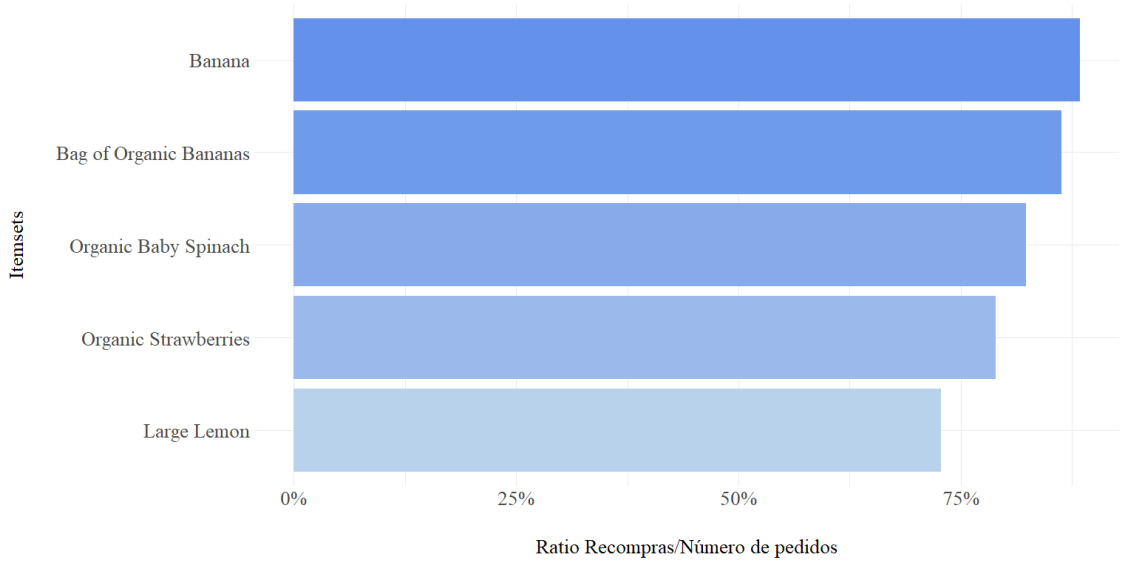
Ahora bien, la Figura 17 expone el ratio de recompras sobre pedidos para los cinco *itemsets* más frecuentes, es decir, el número de veces que se ha recomprado cada uno de estos artículos sobre el total de pedidos en los que aparecen. Es importante señalar que las recompras de cada producto se obtienen a partir de la variable "*reordered*", previamente explicada en la Tabla 13 del apartado 3.

El *itemset* que presenta el ratio más alto es {plátano}, concretamente del 88,41%. Esto significa que dicho *itemset* se ha recomprado en un 88,41% de ocasiones sobre el

número de pedidos, es decir, las recompras del *itemset* {plátano} representan el 88,41% de las compras realizadas del mismo.

Además, cabe destacar que los cinco *itemsets* ofrecen un ratio de recompras sobre pedidos entre el 72,80% y el 88,41%. Esto implica que las recompras representan una proporción de más de la mitad de los pedidos realizados.

Figura 17. Ratio recompras/pedidos de los cinco *itemsets* más frecuentes.



Fuente: elaboración propia con R Studio

Para concluir el análisis de los *itemsets* más frecuentes, a continuación, se estudian los departamentos y pasillos a los que pertenecen y que, por tanto, más se frecuentan. Los cinco conjuntos forman parte del departamento de productos frescos y, como bien mostraba la Figura 12 del apartado 3, el más popular es precisamente dicho departamento.

Por otro lado, los cinco conjuntos más frecuentes se encuentran en el pasillo de frutas frescas, excepto el *itemset* {espinacas pequeñas ecológicas}, que forma parte del pasillo de frutas y verduras envasadas. La Figura 12 ya mostraba que el pasillo más popular era el de frutas frescas. No obstante, también mostraba que el tercero más popular era el de frutas y verduras envasadas, dejando en segundo lugar el pasillo de verduras frescas.

A continuación, se analizan las reglas de asociación identificadas por el algoritmo *a priori* a partir de los *itemsets* frecuentes encontrados. En este caso, se han descubierto **ocho reglas de asociación,** con un nivel mínimo de confianza del 70%.

Por un lado, la Tabla 16 muestra un resumen de las reglas de asociación, donde se identifican los elementos que actúan como antecedentes y consecuentes de las reglas, así como la confianza asociada a cada una de ellas.

Por otro lado, la Figura 18 y la Figura 19 ilustran las reglas de asociación identificadas, destacando en un tono rojo más intenso aquellas con mayor confianza y en un tono rojo más claro aquellas de menor confianza. Sin embargo, la Figura 19 no incluye la primera regla.

Como se puede observar en la Tabla 16, el consecuente de la primera regla {Agua con gas de limón} es antecedente de la tercera y segunda regla, y consecuente de la cuarta regla. Además, el antecedente de la primera regla {Agua con gas de pomelo} es también consecuente de la segunda y tercera regla, y antecedente de la cuarta regla.

No obstante, la Figura 18 revela que la primera regla no se relaciona con ninguna otra. A pesar de que los nombres de ambos elementos en la primera regla tienen el mismo significado que en el resto de las reglas mencionadas, la base de datos los clasifica como productos diferentes. Por ejemplo, se ha observado que el elemento {Agua con gas de limón} en la primera regla posee un identificador distinto al elemento {Agua con gas de limón} presente en la segunda, tercera y cuarta regla. Aun así, cabe destacar que forman parte del mismo departamento y pasillo.

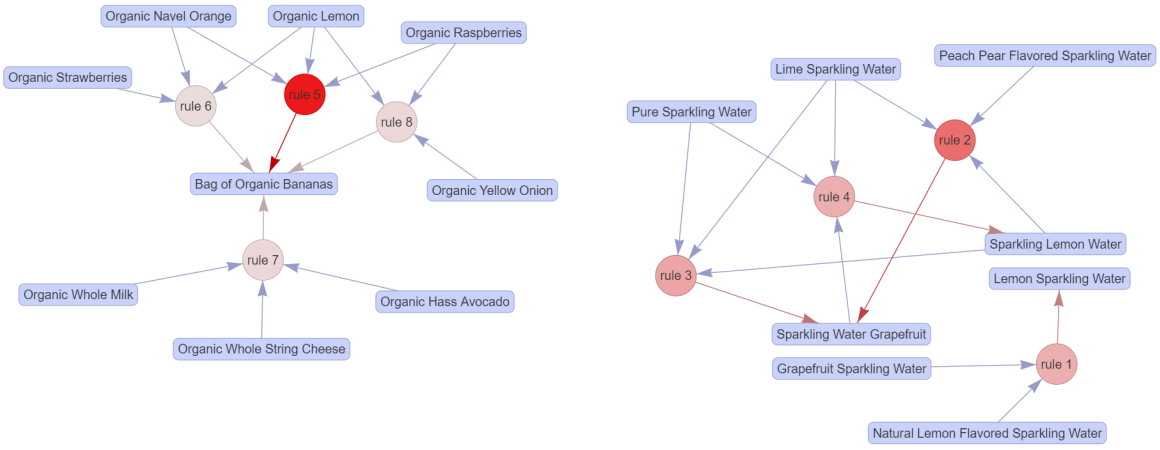
Así pues, solamente se estudiarán las reglas expuestas en la Figura 19, omitiendo la primera regla. No sería adecuado asumir que se trata de los mismos productos, pues la base de datos no los identifica como tal. Asimismo, tampoco sería correcto descartar la posibilidad de que lo fuesen, ya que a la hora de interpretar las reglas resultaría peculiar tratar como productos distintos aquellos que comparten el mismo nombre.

Tabla 16. Resumen de las reglas de asociación identificadas.

Regla	Antecedente	Consecuente	Confianza
1	{Agua con gas de pomelo, Agua con gas natural con sabor a limón}	→ {Agua con gas de limón}	73,84%
2	{Agua con gas de lima, Agua con gas de limón, Agua con gas sabor a pera y melocotón}	→ {Agua con gas de pomelo}	77,78%
3	{Agua con gas de lima, Agua con gas de limón, Agua con gas pura}	→ {Agua con gas de pomelo}	75%
4	{Agua con gas de lima, Agua con gas pura, Agua con gas de pomelo}	→ {Agua con gas de limón}	73,77%
5	{Limón ecológico, Naranja Navel ecológica, Frambuesas ecológicas}	→ {Bolsa de plátanos ecológicos}	81,63%
6	{Limón ecológico, Naranja Navel ecológica, Fresas ecológicas}	→ {Bolsa de plátanos ecológicos}	70,49%
7	{Aguacate Hass ecológico, Leche entera ecológica, Queso entero en tiras ecológico}	→ {Bolsa de plátanos ecológicos}	70,59%
8	{Limón ecológico, Frambuesas ecológicas, Cebolla amarilla ecológica}	→ {Bolsa de plátanos ecológicos}	70,59%

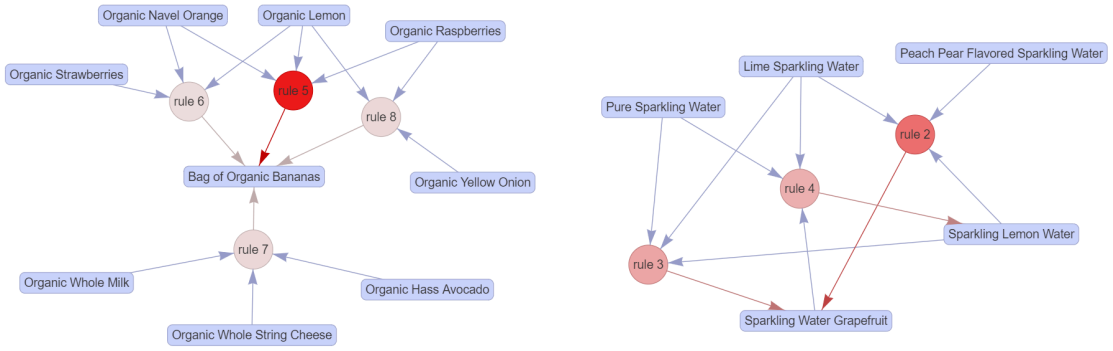
Fuente: elaboración propia con R Studio

Figura 18. Esquema de las reglas de asociación identificadas.



Fuente: elaboración propia con R Studio

Figura 19. Esquema de las reglas de asociación identificadas (omitiendo la primera regla).



Fuente: elaboración propia con R Studio

Por un lado, la Figura 19 ilustra la relación entre la segunda, tercera y cuarta regla. Como se puede observar, los productos implicados en todas ellas son diversos sabores de agua con gas. A continuación, se detallan cada una de estas reglas:

- **Segunda regla:** existe una probabilidad del 77,78% de que, al comprar conjuntamente agua con gas de lima, agua con gas de limón y agua con gas sabor a pera y melocotón, también se compre agua con gas de pomelo.

- **Tercera regla:** existe una probabilidad del 75% de que, comprando de manera conjunta agua con gas de lima, agua con gas de limón y agua con gas pura, también se compre agua con gas de pomelo.
- **Cuarta regla:** existe una probabilidad del 73,77% de que, al comprar simultáneamente agua con gas de lima, agua con gas pura, y agua con gas de pomelo, también se compre agua con gas de limón.

Estas reglas evidencian una interconexión entre los diferentes productos. Por ejemplo, la segunda y tercera regla indican una alta probabilidad de que la compra de determinados sabores de agua con gas implique la compra de agua con gas de pomelo. Asimismo, la cuarta regla sugiere que existe una alta probabilidad de que comprando agua con gas de pomelo junto con agua con gas pura y agua con gas de lima, también se compre agua con gas de limón.

Nótese que el artículo agua con gas de limón no es solamente resultado de la cuarta regla sino también antecedente tanto de la segunda como de la tercera regla, demostrando así la interconexión explicada.

Por otro lado, la Figura 19 ilustra la relación entre la quinta, sexta, séptima y octava regla. Cada una de estas reglas se explican a continuación:

- **Quinta regla:** existe una probabilidad del 81,63% de que, al comprar conjuntamente limón ecológico, naranja Navel orgánica y frambuesas ecológicas, también se compre una bolsa de plátanos ecológicos.
- **Sexta regla:** existe una probabilidad del 70,49% de que, comprando de manera conjunta limón ecológico, naranja Navel orgánica y fresas ecológicas, también se compre una bolsa de plátanos ecológicos.
- **Séptima regla:** existe una probabilidad del 70,59% de que, al comprar simultáneamente aguacate Hass orgánico, leche entera ecológica y queso entero en tiras ecológico, también se compre una bolsa de plátanos ecológicos.
- **Octava regla:** existe una probabilidad del 70,59% de que, comprando simultáneamente limón ecológico, frambuesas ecológicas, cebolla amarilla ecológica, también se compre una bolsa de plátanos ecológicos.

Es importante destacar que el elemento consecuente de todas estas reglas es {bolsa de plátanos ecológicos}. Sin embargo, este elemento no actúa como antecedente en ninguna otra regla, lo que significa que su compra no influye de manera significativa en la compra de otros productos. No obstante, el hecho de comprar una bolsa de plátanos ecológicos puede estar determinado por el hecho de comprar ciertos productos, concretamente aquellos implicados en estas reglas.

Además, se destaca que los productos naranja Navel orgánica, limón ecológico y frambuesas ecológicas son antecedentes de la quinta, sexta y octava regla, mientras que el producto fresas ecológicas solo es antecedente de la sexta regla. A pesar de que la séptima regla comparte el mismo resultado que las reglas quinta, sexta y octava, sus antecedentes no se relacionan con el resto de las reglas.

Finalmente, la Tabla 17 muestra las siguientes **métricas de evaluación** para cada una de las reglas de asociación encontradas: soporte, cobertura, *lift*, y Chi-cuadrado. Además, se analiza la relación entre el soporte y la cobertura de cada regla de asociación, es decir, dada una regla $A \rightarrow B$, se examina la frecuencia con la que aparecen conjuntamente A y B con respecto a la frecuencia con la que aparece A.

Comparando la Tabla 16 y la Tabla 17, se observa que la relación entre el soporte y la cobertura es equivalente a la confianza. La regla de mayor confianza es la quinta, con un nivel del 81,63%. Las reglas segunda, tercera y cuarta presentan niveles de confianza que rondan aproximadamente entre el 73% y el 77%, mientras que las reglas sexta, séptima y octava exhiben los niveles de confianza más bajos, sin llegar a alcanzar el 71%.

Resulta relevante señalar que la octava regla destaca por tener el mayor soporte y la mayor cobertura, aunque esto da lugar a uno de los menores niveles de confianza. A pesar de que todos los productos involucrados en esta regla aparecen juntos en más transacciones en comparación al resto de reglas, esta frecuencia solamente representa un 70,59% de las transacciones en las que aparecen de manera conjunta los elementos antecedentes de la regla.

Asimismo, la quinta regla cuenta con la segunda menor cobertura y el segundo menor nivel de soporte, resultando en el mayor nivel de confianza. Aunque los artículos involucrados en la quinta regla aparecen juntos en menos transacciones en comparación

al resto de reglas, en este caso, esta frecuencia logra representar el 81,63% de las transacciones en las que aparecen conjuntamente los elementos antecedentes de la regla.

Por otro lado, cabe destacar que todas las reglas de asociación cuentan con un *lift* superior a uno, lo que significa que el conjunto antecedente y el consecuente no son estadísticamente independientes. Por ejemplo, la cuarta regla tiene un *lift* de 68,02, lo que significa que la probabilidad de que todos los elementos involucrados en la regla se compren de manera conjunta es 68,02 veces la probabilidad de que se compren de forma independiente.

Como se puede observar, desde la quinta hasta la octava regla, el *lift* es más bajo en comparación al resto de reglas. Esto indica que la relación entre el conjunto antecedente y el consecuente de estas reglas no es tan fuerte como en el resto.

La prueba estadística Chi-cuadrado también estudia la relación de independencia de los conjuntos antecedente y consecuente de las reglas. Como se puede observar, el estadístico Chi-cuadrado es mayor que 169 en todos los casos, lo que indica que se rechaza la hipótesis nula de independencia entre ambos conjuntos y, por consiguiente, se infiere una relación significativa y fuerte entre estos conjuntos.

Tabla 17. Medidas de evaluación de las reglas de asociación.

Regla	Soporte	Cobertura	Soporte/Cobertura	*Lift*	Chi-cuadrado
1	0,037%	0,049%	0,7384	210,64	10054,99
2	0,032%	0,041%	0,7778	30,38	1225,26
3	0,034%	0,046%	0,7500	29,30	1262,77
4	0,034%	0,047%	0,7377	68,02	3005,57
5	0,030%	0,037%	0,8163	6,91	229,73
6	0,033%	0,046%	0,7049	5,97	202,04
7	0,027%	0,039%	0,7059	5,98	169,46
8	0,037%	0,052%	0,7059	5,98	225,97

Fuente: elaboración propia con R Studio

5. Conclusiones.

El presente trabajo se centra en el análisis de datos de transacciones de la plataforma Instacart con el fin de descubrir patrones de compra y relaciones entre productos que puedan mejorar las estrategias comerciales y la experiencia del usuario en el comercio electrónico. A través de este análisis, se han cumplido los objetivos planteados al inicio de la investigación, lo que se refleja en las siguientes conclusiones.

En primer lugar, se ha logrado identificar una serie de artículos frecuentes, como plátanos, espinacas, fresas y limones, entre otros. Estos artículos destacan no solo por ser los más populares entre los usuarios, sino también por tratarse de productos frescos, concretamente frutas y verduras frescas.

La frecuencia de compra de estos productos sugiere una creciente preferencia por opciones más saludables y ecológicas. Esta tendencia puede tener un impacto significativo en las estrategias de marketing de Instacart, así como en la selección de productos ofrecidos en su plataforma. Es importante que la empresa reconozca esta tendencia y adapte sus estrategias para satisfacer las demandas cambiantes de los consumidores. Por ejemplo, podría implicar aumentar la disponibilidad de productos ecológicos en su plataforma.

Asimismo, se observa una destacada fidelidad de los consumidores hacia los productos mencionados, reflejada en su alta tasa de recompra. Este comportamiento sugiere una clara satisfacción por parte de los usuarios de Instacart con ciertos artículos, lo cual podría influir en las decisiones relacionadas con la promoción de estos productos.

En cuanto a los días de la semana más frecuentes para realizar compras, se ha confirmado que los fines de semana son los momentos de mayor actividad, lo que coincide con la tendencia general del comercio *online*. Esta información puede ser útil para planificar campañas promocionales y ofertas especiales en momentos de alta demanda.

Por otro lado, el análisis de las reglas encontradas ha revelado patrones interesantes en los usuarios, específicamente en la compra de plátanos y agua con gas. La alta frecuencia de compra de plátanos y su relación con otros artículos señala su importancia como producto básico entre los usuarios.

Del mismo modo, el patrón de compra de diversos sabores de agua con gas sugiere la disposición de los consumidores a experimentar, lo que abre oportunidades para introducir nuevos productos e implementar estrategias de comercialización.

Las reglas de asociación identificadas serán fundamentales en las mejoras del diseño y la estructura de la plataforma de Instacart, así como en la personalización de recomendaciones para los usuarios. La alta confianza y relevancia de algunas de estas reglas destacan la existencia de patrones claros en el comportamiento de los consumidores.

Así pues, los hallazgos de esta investigación ofrecen importantes *insights* para Instacart, proporcionando una comprensión más profunda del comportamiento de compra en el entorno digital. Estos *insights* pueden ser utilizados para desarrollar estrategias comerciales más efectivas, mejorando la experiencia del usuario y aumentando la satisfacción del cliente.

No obstante, se podría dar un paso más allá en este estudio mediante la implementación de técnicas más avanzadas para la búsqueda de reglas de asociación, como el algoritmo *Frequent Pattern* (FP) *growth*. Asimismo, sería interesante evaluar el potencial impacto que nuevas estrategias comerciales, basadas en los patrones encontrados, podrían llegar a tener en los resultados financieros de Instacart.

6. Declaración de uso de herramientas de IA generativa.

Por la presente, yo, Lucía Gema Hernando García, estudiante de ADE y Business Analytics de la Universidad Pontificia Comillas al presentar mi Trabajo Fin de Grado titulado "Análisis de la cesta de la compra en Instacart: extracción de patrones de consumo implementando el algoritmo *a priori*", declaro que he utilizado la herramienta de Inteligencia Artificial Generativa ChatGPT u otras similares de IAG de código sólo en el contexto de las actividades descritas a continuación:

1. Interpretador de código: Para realizar análisis de datos preliminares.
2. Corrector de estilo literario y de lenguaje: Para mejorar la calidad lingüística y estilística del texto.
3. Traductor: Para traducir textos de un lenguaje a otro.

Afirmo que toda la información y contenido presentados en este trabajo son producto de mi investigación y esfuerzo individual, excepto donde se ha indicado lo contrario y se han dado los créditos correspondientes (he incluido las referencias adecuadas en el TFG y he explicitado para que se ha usado ChatGPT u otras herramientas similares). Soy consciente de las implicaciones académicas y éticas de presentar un trabajo no original y acepto las consecuencias de cualquier violación a esta declaración.

Fecha: 16 de febrero de 2024

Firma: Lucía Gema Hernando García

7. Bibliografía.

Beltrán Martínez, B. (2001). Minería de datos. Benemérita Universidad Autónoma de Puebla. https://www.cs.buap.mx/~bbeltran/NotasMD.pdf

Castro Casillas, G. (2016). *Uso de análisis asociativo en algoritmos de aprendizaje.* http://hdl.handle.net/10486/670712

Chaudhuri, N., Gupta, G., Vamsi, V., & Bose, I. (2021). On the platform but will they buy? predicting customers' purchase behavior using Deep Learning. *Decision Support Systems*, *149*, 113622. https://doi.org/10.1016/j.dss.2021.113622

Díaz Llerena, L. (2013). *Implementación de un algoritmo Apriori-like-1 para el minado de reglas de asociación difusas.* https://repositorio.uci.cu/jspui/handle/ident/8712

Díaz-Molina, A., & García-Garrido, L. (2018). FP-MAXFLOW: Un algoritmo para la minería de patrones relevantes de longitud máxima. *Computación y Sistemas*, *22*(2), 563-583.

Hahsler, M. (2015). A Probabilistic Comparison of Commonly Used Interest Measures for Association Rules. *United States*. *Southern Methodist University*, *27*. https://mhahsler.github.io/arules/docs/measures

Isinkaye, F. O., Folajimi, Y. O., & Ojokoh, B. A. (2015). Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, *16*(3), 261–273. https://doi.org/10.1016/j.eij.2015.06.005

Lunde, E. K., & JIANG, X. (2023). *How does dynamic pricing affect customer perceived fairness across product necessities?* (Master's thesis, Handelshøyskolen BI).

Malberti Riveros, M. A., & Elida Beguerí, G. (2015). Reglas de Asociación con los datos de una biblioteca universitaria. *Revista Cubana de Ciencias Informáticas, 9*(4), 30-45.

Moya Amaris, M. E., y Rodríguez Rodríguez, J. E. (2003). La contribución de las reglas de asociación a la minería de datos. *Tecnura*, *7*(13), 94–109.

Narvekar, M., & Syed, S. F. (2015). An optimized algorithm for association rule mining using FP Tree. *Procedia Computer Science*, *45*, 101–110. https://doi.org/10.1016/j.procs.2015.03.097

Orozco Bohórquez, M. (2017). *Método de reglas de asociación para el análisis de afinidad entre objetos de tipo texto*. Repositorio CUC. http://hdl.handle.net/11323/165

Psparks. (2017). *Instacart Market Basket Analysis*. Kaggle. https://www.kaggle.com/datasets/psparks/instacart-market-basket-analysis?select=orders.csv

Rao, S., & Zhang, L. (2021). The algorithms that make Instacart Roll: How Machine Learning and Other Tech Tools Guide your groceries from store to doorstep. *IEEE Spectrum*, *58*(3), 36–42. https://doi.org/10.1109/mspec.2021.9370062

Riquelme Santos, J. C., Ruiz, R., & Gilbert, K. (2006). Minería de datos: Conceptos y tendencias. *Inteligencia Artificial: Revista Iberoamericana de Inteligencia Artificial*, *10* (29), 11-18.

Sáenz López, A., Cortés Martínez, F., & Betancourt Chávez, J. R. (2017). Reglas de asociación en una Base de datos del área médica. *Revista Arquitectura e Ingeniería*, *11*(2), 5.

Verastegui Tene, F., & Vargas Merino, J. (2021). Estrategias de merchandising: un análisis de su efectividad para la atracción de nuevos clientes. *RAN. Revista Academia y Negocios*, *7*(1), 41–54. https://doi.org/10.29393/ran6-4emfv20004

8. Anexo I: código.

```
######################### CARGA DE LIBRERÍAS ##########################

library(tidyverse)
library(readr)
library(arules)
library(ggplot2)
library(reshape2)
library(scales)
library(gridExtra)
library(arulesViz)


######################### IMPORTACIÓN DE DATOS ##########################

aisles <- read_csv("aisles.csv")
departments <- read_csv("departments.csv")
orders <- read_csv("orders.csv")
products <- read_csv("products.csv")
op_prior <- read_csv("order_products__prior.csv")
op_train <- read_csv("order_products__train.csv")


############## UNIÓN BASES DE DATOS ANÁLISIS EXPLORATORIO #############

# 'data_an_expl' une los productos con sus pasillos y departamentos
data_an_expl <- merge(products, aisles, by="aisle_id")
data_an_expl <- merge(data_an_expl, departments, by="department_id")

# Se  une a 'data_an_expl' el dataset 'op_prior'
data_an_expl <- merge(data_an_expl, op_prior, by="product_id")

# Se  une a 'data_an_expl' el dataset 'orders'
data_an_expl <- merge(data_an_expl, orders, by="order_id")


############### UNIÓN BASES DE DATOS ALGORITMO A PRIORI ###############

# 'data_apriori' une 'op_train' y 'products'
data_apriori <- merge(op_train, products, by="product_id")

# 'data_apriori' se queda solo con las variables necesarias
data_apriori <- data.frame("order_id" = data_apriori$order_id,
                           "product_name" = data_apriori$product_name)

# 'data_graf_apriori' se crea para un análisis posterior al algoritmo
data_graf_apriori <- merge(products, aisles, by="aisle_id")
data_graf_apriori <- merge(data_graf_apriori, departments,
by="department_id")
```

```
data_graf_apriori <- merge(data_graf_apriori, op_train,
by="product_id")
data_graf_apriori <- merge(data_graf_apriori, orders, by="order_id")


######################## ANÁLISIS EXPLORATORIO ########################

# Figura 5: Distribución de pedidos por cliente
orders_per_user <- data_an_expl %>%
  group_by(user_id) %>%
  summarise(num_orders = n_distinct(order_id))

ggplot(orders_per_user, aes(x = num_orders)) +
  geom_histogram(binwidth = 3,
                 fill = 'steelblue2',
                 color = 'black') +
  geom_density(aes(y = ..count..), color = 'brown3') +
  labs(x = 'Número de pedidos',
       y = 'Número de clientes') +
  scale_x_continuous(labels = number_format(
    scale = 1,
    big.mark = ".",
    decimal.mark = ","
  )) +
  scale_y_continuous(labels = number_format(
    scale = 1,
    big.mark = ".",
    decimal.mark = ","
  )) +
  theme_minimal() +
  theme(
    text = element_text(family = "serif", size = 20),
    axis.text = element_text(size = 20),
    axis.title.y = element_text(margin = margin(r = 30)),
    axis.title.x = element_text(margin = margin(t = 30))
  )


# Figura 6: Número de pedidos por día de la semana
orders_per_dow <- data_an_expl %>%
  group_by(order_dow) %>%
  summarise(num_orders = n_distinct(order_id))

ggplot(data = orders_per_dow, aes(
  x = factor(
    order_dow,
    labels = c("Dom", "Lun", "Mar", "Mié", "Jue", "Vie", "Sáb")
  ),
  y = num_orders,
  fill = num_orders
)) +
  geom_bar(stat = "identity") +
```

```
  scale_fill_gradient(
    low = "chocolate2",
    high = "palegreen3",
    labels = number_format(
      scale = 1,
      big.mark = ".",
      decimal.mark = ","
    )
  ) +
  labs(x = 'Día de la semana',
       y = 'Número de pedidos',
       fill = 'Número de pedidos') +
  scale_y_continuous(labels = number_format(
    scale = 1,
    big.mark = ".",
    decimal.mark = ","
  )) +
  theme_minimal() +
  theme(
    text = element_text(family = "serif", size = 20),
    axis.text = element_text(size = 20),
    axis.title.y = element_text(margin = margin(r = 30)),
    axis.title.x = element_text(margin = margin(t = 30))
  )


# Figura 7: Número de pedidos por hora del día
orders_per_hour <- data_an_expl %>%
  group_by(order_hour_of_day) %>%
  summarise(num_orders = n_distinct(order_id))

ggplot(data = orders_per_hour,
       aes(x = order_hour_of_day, y = num_orders, fill = num_orders))
+
  geom_bar(stat = "identity") +
  scale_fill_gradient(
    low = "chocolate2",
    high = "palegreen3",
    labels = number_format(
      scale = 1,
      big.mark = ".",
      decimal.mark = ","
    )
  ) +
  labs(x = 'Hora',
       y = 'Número de pedidos',
       fill = 'Número de pedidos') +
  scale_y_continuous(labels = number_format(
    scale = 1,
    big.mark = ".",
    decimal.mark = ","
  )) +
```

```
  theme_minimal() +
  theme(
    text = element_text(family = "serif", size = 20),
    axis.text = element_text(size = 20),
    axis.title.y = element_text(margin = margin(r = 30)),
    axis.title.x = element_text(margin = margin(t = 30))
  )


# Figura 8: Número de productos comprados por día y hora
days_hours <- data_an_expl %>%
  group_by(order_dow, order_hour_of_day) %>%
  summarize(order_number = n())

days_hours <-
  dcast(days_hours, order_hour_of_day ~ order_dow, value.var =
"order_number")

ggplot(data = melt(days_hours),
       aes(
         x = order_hour_of_day,
         y = factor(
           variable,
           labels = c("Dom", "Lun", "Mar", "Mié", "Jue", "Vie", "Sáb")
         ),
         fill = value
       )) +
  geom_tile() +
  scale_fill_gradient(
    low = "gray100",
    high = "steelblue2",
    labels = number_format(
      scale = 1,
      big.mark = ".",
      decimal.mark = ","
    )
  ) +
  labs(x = 'Hora',
       y = 'Día de la semana',
       fill = 'Productos comprados') +
  theme_minimal() +
  theme(
    text = element_text(family = "serif", size = 20),
    axis.text = element_text(size = 20),
    axis.title.y = element_text(margin = margin(r = 30)),
    axis.title.x = element_text(margin = margin(t = 30))
  )
```

```r
# Figura 9: Pedidos según el número de días desde el último pedido
orders_priororder <- data_an_expl %>%
  group_by(days_since_prior_order) %>%
  summarise(num_orders = n_distinct(order_id))

ggplot(data = orders_priororder,
       aes(x = days_since_prior_order, y = num_orders, fill =
num_orders)) +
  geom_bar(stat = "identity") +
  scale_fill_gradient(
    low = "chocolate2",
    high = "palegreen3",
    labels = number_format(
      scale = 1,
      big.mark = ".",
      decimal.mark = ","
    )
  ) +
  labs(x = 'Número de días',
       y = 'Número de pedidos',
       fill = 'Número de pedidos') +
  scale_x_continuous(labels = number_format(
    scale = 1,
    big.mark = ".",
    decimal.mark = ","
  )) +
  scale_y_continuous(labels = number_format(
    scale = 1,
    big.mark = ".",
    decimal.mark = ","
  )) +
  theme_minimal() +
  theme(
    text = element_text(family = "serif", size = 20),
    axis.text = element_text(size = 20),
    axis.title.y = element_text(margin = margin(r = 30)),
    axis.title.x = element_text(margin = margin(t = 30))
  )


# Figura 10: Artículos populares
product_frequencies <- data_an_expl %>%
  group_by(product_name) %>%
  summarize(order_frequency = n_distinct(order_id)) %>%
  arrange(desc(order_frequency)) %>%
  head(12)

comp <-
  ggplot(data = product_frequencies,
         aes(
           x = order_frequency,
           y = reorder(product_name, order_frequency),
```

```
          fill = order_frequency
        )) +
  geom_bar(stat = "identity") +
  scale_fill_gradient(
    low = "chocolate2",
    high = "palegreen3",
    labels = number_format(
      scale = 1,
      big.mark = ".",
      decimal.mark = ","
    )
  ) +
  labs(x = 'Número de pedidos',
       y = 'Productos',
       fill = 'Número de pedidos') +
  ggtitle("Productos más comprados") +
  guides(fill = "none") +
  scale_x_continuous(labels = number_format(
    scale = 1,
    big.mark = ".",
    decimal.mark = ","
  )) +
  theme_minimal() +
  theme(
    text = element_text(family = "serif", size = 20),
    axis.text.y = element_text(size = 20),
    axis.text.x = element_text(size = 18, hjust= 1, angle = 45),
    axis.title.y = element_text(margin = margin(r = 30)),
    axis.title.x = element_text(margin = margin(t = 30))
  )

reordered_products <- data_an_expl %>%
  group_by(product_name) %>%
  summarize(reordered_count = sum(reordered)) %>%
  arrange(desc(reordered_count)) %>%
  head(12)

recomp <-
  ggplot(data = reordered_products,
         aes(
           x = reordered_count,
           y = reorder(product_name, reordered_count),
           fill = reordered_count
         )) +
  geom_bar(stat = "identity") +
  scale_fill_gradient(
    low = "chocolate2",
    high = "palegreen3",
    labels = number_format(
      scale = 1,
      big.mark = ".",
      decimal.mark = ","
```

```
    )
  ) +
  labs(x = 'Número de recompras',
       y = 'Productos',
       fill = 'Número de recompras') +
  ggtitle("Productos más recomprados") +
  guides(fill = "none") +
  scale_x_continuous(labels = number_format(
    scale = 1,
    big.mark = ".",
    decimal.mark = ","
  )) +
  theme_minimal() +
  theme(
    text = element_text(family = "serif", size = 20),
    axis.text.y = element_text(size = 20),
    axis.text.x = element_text(size = 18, hjust= 1, angle = 45),
    axis.title.y = element_blank(),
    axis.title.x = element_text(margin = margin(t = 30))
  )

art_populares <- grid.arrange(comp, recomp, ncol = 2)


# Figura 11: Distribución de pedidos en función de su tamaño
products_per_order <- data_an_expl %>%
  group_by(order_id) %>%
  summarize(number_products = max(add_to_cart_order))

ggplot(data = products_per_order, aes(x = number_products)) +
  geom_histogram(binwidth = 1,
                 fill = "steelblue2",
                 color = "black") +
  labs(x = "Número de productos",
       y = "Número de pedidos") +
  scale_x_continuous(labels = number_format(
    scale = 1,
    big.mark = ".",
    decimal.mark = ","
  )) +
  scale_y_continuous(labels = number_format(
    scale = 1,
    big.mark = ".",
    decimal.mark = ","
  )) +
  theme_minimal() +
  theme(
    text = element_text(family = "serif", size = 20),
    axis.text = element_text(size = 20),
    axis.title.y = element_text(margin = margin(r = 30)),
    axis.title.x = element_text(margin = margin(t = 30))
  )
```

```r
# Figura 12: Pasillos y departamentos más populares
top_aisles <- data_an_expl %>%
  group_by(aisle) %>%
  summarize(number_orders = n_distinct(order_id)) %>%
  arrange(desc(number_orders)) %>%
  head(10)

pasillos <-
  ggplot(data = top_aisles, aes(
    x = number_orders,
    y = reorder(aisle, number_orders),
    fill = number_orders
  )) +
  geom_bar(stat = "identity") +
  scale_fill_gradient(
    low = "chocolate2",
    high = "palegreen3",
    labels = number_format(
      scale = 1,
      big.mark = ".",
      decimal.mark = ","
    )
  ) +
  labs(x = 'Número de pedidos',
       y = 'Pasillos',
       fill = 'Número de pedidos') +
  ggtitle("Pasillos más populares") +
  guides(fill = "none") +
  scale_x_continuous(labels = number_format(
    scale = 1,
    big.mark = ".",
    decimal.mark = ","
  )) +
  theme_minimal() +
  theme(
    text = element_text(family = "serif", size = 20),
    axis.text.y = element_text(size = 20),
    axis.text.x = element_text(
      size = 18,
      hjust = 1,
      angle = 45
    ),
    axis.title.y = element_text(margin = margin(r = 20)),
    axis.title.x = element_text(margin = margin(t = 20))
  )

top_departments <- data_an_expl %>%
  group_by(department) %>%
  summarize(number_orders = n_distinct(order_id)) %>%
  arrange(desc(number_orders)) %>%
  head(10)
```

```
depart <-
  ggplot(data = top_departments, aes(
    x = number_orders,
    y = reorder(department, number_orders),
    fill = number_orders
  )) +
  geom_bar(stat = "identity") +
  scale_fill_gradient(
    low = "chocolate2",
    high = "palegreen3",
    labels = number_format(
      scale = 1,
      big.mark = ".",
      decimal.mark = ","
    )
  ) +
  labs(x = 'Número de pedidos',
       y = 'Departamentos',
       fill = 'Número de pedidos') +
  ggtitle("Departamentos más populares") +
  guides(fill = "none") +
  scale_x_continuous(labels = number_format(
    scale = 1,
    big.mark = ".",
    decimal.mark = ","
  )) +
  theme_minimal() +
  theme(
    text = element_text(family = "serif", size = 20),
    axis.text.y = element_text(size = 20),
    axis.text.x = element_text(
      size = 18,
      hjust = 1,
      angle = 45
    ),
    axis.title.y = element_text(margin = margin(r = 20)),
    axis.title.x = element_text(margin = margin(t = 20))
  )

pas_dep <- grid.arrange(pasillos, depart, ncol = 2)


#################### APLICACIÓN ALGORITMO A PRIORI ####################

# Se transforma el dataset a la forma de "transacciones"
data_apriori <-
  split(x = data_apriori$product_name, f = data_apriori$order_id)
transacciones <- as(data_apriori, Class = "transactions")

# Se inspeccionan las transacciones
inspect(transacciones[1:5])
```

```r
# Se analiza el tamaño de las transacciones
sizetran <- size(transacciones)
summary(sizetran)

# Figura 13: Percentiles del tamaño de las transacciones
quantiles <- quantile(sizetran, probs = seq(0, 1, 0.1))
df_quantiles <-
  data.frame(Percentiles = names(quantiles),
             Valor = as.numeric(quantiles))
df_quantiles$Percentiles <-
  factor(df_quantiles$Percentiles, levels = names(quantiles))

ggplot(df_quantiles, aes(x = Percentiles, y = Valor)) +
  geom_point(color = "red", size = 4) +
  labs(x = "Percentil", y = "Número de artículos") +
  theme_minimal() +
  theme(
    text = element_text(family = "serif", size = 20),
    axis.text = element_text(size = 20),
    axis.title.y = element_text(margin = margin(r = 30)),
    axis.title.x = element_text(margin = margin(t = 30))
  )

# Se obtiene el soporte mínimo
frecuencia_relat <-
  itemFrequency(x = transacciones, type = "relative")
frecuencia_relat %>% sort(decreasing = TRUE) %>% head(5)

soporte <- mean(frecuencia_relat)

# PASO 1: Itemsets frecuentes

itemsets <- apriori(
  data = transacciones,
  parameter = list(
    support = soporte,
    confidence = 0.70,
    maxlen = dim(transacciones)[1],
    maxtime = 60,
    target = "frequent itemset"
  )
)

summary(itemsets)

# PASO 2: Reglas de asociación

reglas <- apriori(
  data = transacciones,
  parameter = list(
    support = soporte,
    confidence = 0.70,
```

```
    maxlen = dim(transacciones)[1],
    maxtime = 60,
    target = "rules"
  )
)
summary(reglas)


#################### RESULTADOS ALGORITMO A PRIORI ####################

# Figura 14:Tamaño de los itemsets identificados
tamaños <- c(1, 2, 3, 4, 5)
frecuencias <- c(5949, 16030, 8044, 787, 8)
datos_linea <-
  data.frame(Tamaño = tamaños, Frecuencia = frecuencias)

ggplot(datos_linea, aes(x = Tamaño, y = Frecuencia)) +
  geom_line(color = "dodgerblue", size = 1.5) +
  geom_point(color = "dodgerblue", size = 5) +
  labs(x = "Tamaño (nº elementos)", y = "Itemsets") +
  theme_minimal() +
  scale_y_continuous(labels = label_number(big.mark = ".",
decimal.mark = ",")) +
  theme(
    text = element_text(family = "serif", size = 20),
    axis.text = element_text(size = 20),
    axis.title.y = element_text(margin = margin(r = 30)),
    axis.title.x = element_text(margin = margin(t = 30))
  )


# Figura 15: Los 20 itemsets más frecuentes
top_20_itemsets <-
  sort(itemsets, by = "support", decreasing = TRUE)[1:20]
inspect(top_20_itemsets)

as(top_20_itemsets, Class = "data.frame") %>%
  ggplot(aes(
    x = reorder(items, support),
    y = support,
    fill = support
  )) +
  geom_col() +
  scale_fill_gradient(low = "slategray2",
                      high = "cornflowerblue",
                      labels = scales::comma) +
  coord_flip() +
  labs(x = "Itemsets", y = "Soporte", fill = "Soporte") +
  theme(
    text = element_text(family = "serif", size = 20),
    axis.text = element_text(size = 20),
    axis.title.y = element_text(margin = margin(r = 30)),
```

```
    axis.title.x = element_text(margin = margin(t = 30))
  )

inspect(sort(itemsets[size(itemsets) > 1], decreasing = TRUE)[1:20])

# Filtro artículos más frecuentes para las próximas gráficas
mas_frecuentes <- data_graf_apriori %>%
  filter(
    product_name %in% c(
      "Banana",
      "Bag of Organic Bananas",
      "Organic Strawberries",
      "Organic Baby Spinach",
      "Large Lemon"
    )
  )

# Figura 16: Número de pedidos por día de la semana
pedidos_por_dia_producto <- mas_frecuentes %>%
  group_by(order_dow, product_name) %>%
  summarise(num_pedidos = n()) %>%
  ungroup()

pedidos_totales_por_dia <- data_graf_apriori %>%
  group_by(order_dow) %>%
  summarise(total_pedidos = n_distinct(order_id)) %>%
  ungroup()

porcentaje_pedidos <- pedidos_por_dia_producto %>%
  inner_join(pedidos_totales_por_dia, by = "order_dow") %>%
  mutate(percentage = (num_pedidos / total_pedidos))

colores <-
  c(
    "Banana" = "tan2",
    "Bag of Organic Bananas" = "lightgoldenrod",
    "Organic Strawberries" = "mediumseagreen",
    "Organic Baby Spinach" = "skyblue1",
    "Large Lemon" = "rosybrown2"
  )

ggplot(porcentaje_pedidos,
       aes(x = order_dow, y = percentage, fill = product_name)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(x = "Días de la semana",
       y = "Pedidos (%)",
       fill = "Itemsets") +
  scale_x_continuous(
    breaks = 0:6,
    labels = c("Dom", "Lun", "Mar", "Mié", "Jue", "Vie", "Sáb")
  ) +
  scale_y_continuous(labels = percent_format()) +
```

```
  scale_fill_manual(values = colores) +
  theme_minimal() +
  theme(
    text = element_text(family = "serif", size = 20),
    axis.text = element_text(size = 20),
    axis.title.y = element_text(margin = margin(r = 30)),
    axis.title.x = element_text(margin = margin(t = 30))
  )


# Figura 17: Ratio recompras/pedidos
recompras <- mas_frecuentes %>%
  filter(reordered == 1) %>%
  group_by(product_name) %>%
  summarise(recompras = n()) %>%
  ungroup()

pedidos_por_producto <- mas_frecuentes %>%
  group_by(product_name) %>%
  summarise(num_pedidos = n_distinct(order_id))

ratio_rec_ped <- recompras %>%
  inner_join(pedidos_por_producto, by = "product_name") %>%
  mutate(ratio = (recompras / num_pedidos))

ggplot(ratio_rec_ped, aes(
  x = reorder(product_name, ratio),
  y = ratio,
  fill = ratio
)) +
  geom_bar(stat = "identity") +
  labs(x = "Itemsets", y = "Ratio Recompras/Número de pedidos") +
  theme_minimal() +
  scale_y_continuous(labels = label_number(big.mark = ".",
decimal.mark = ",")) +
  scale_fill_gradient(low = "slategray2", high = "cornflowerblue") +
  scale_y_continuous(labels = percent_format()) +
  guides(fill = "none") +
  coord_flip() +
  theme(
    text = element_text(family = "serif", size = 20),
    axis.text = element_text(size = 20),
    axis.title.y = element_text(margin = margin(r = 30)),
    axis.title.x = element_text(margin = margin(t = 30))
  )
```

```r
# Figuras 18 y 19: Esquema de las reglas de asociación identificada
plot(
  reglas,
  method = "graph",
  measure = "confidence",
  shading = "confidence",
  engine = "html"
)

# Métricas de evalución de las reglas
metricas <-
  interestMeasure(
    reglas,
    measure = c("coverage", "support", "lift", "chiSquared"),
    transactions = transacciones
  )
metricas

cob_sop <- metricas$coverage / metricas$support
quality(reglas) <- cbind(quality(reglas), metricas, cob_sop)

inspect(x = reglas)
df_reglas <- as(reglas, Class = "data.frame")
df_reglas %>% as_tibble() %>% arrange(desc(confidence)) %>% head()
```