

UNIVERSIDAD POLITÉCNICA DEL VALLE DE TOLUCA

NOMBRE DEL PROGRAMA EDUCATIVO:
INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN.

NOMBRE DE LA ASIGNATURA:
INTELIGENCIA DE NEGOCIOS

NOMBRE DE LA PRÁCTICA:
PROYECTO U3

NOMBRE DE LA UNIDAD DE APRENDIZAJE:
UNIDAD: III. MINERÍA DE DATOS

FACILITADOR:
MTRO. LEDEZMA SÁNCHEZ JOSÉ LUIS

INTEGRANTES:

No	Matrícula	Nombre	Grupo
1	1320343029	COLIN VIEYRA JONATHAN IVAN	ITI9MA
2	1319313066	MENDOZA GARCÍA EMMANUEL	ITI9MA

JULIO 2024.

Introducción.

La minería de datos se ha considerado una alternativa para dar solución a diferentes problemáticas que requieren el análisis de grandes volúmenes de datos. En esta presentación se abarcará el tema de las asociaciones Agrawal (Algoritmo Apriori). Las reglas de asociación fueron utilizadas para el análisis de las cestas de mercado, a partir de dicho estudio se logró establecer criterios para tomar decisiones en base al comportamiento de compra de los productos por parte de los clientes.

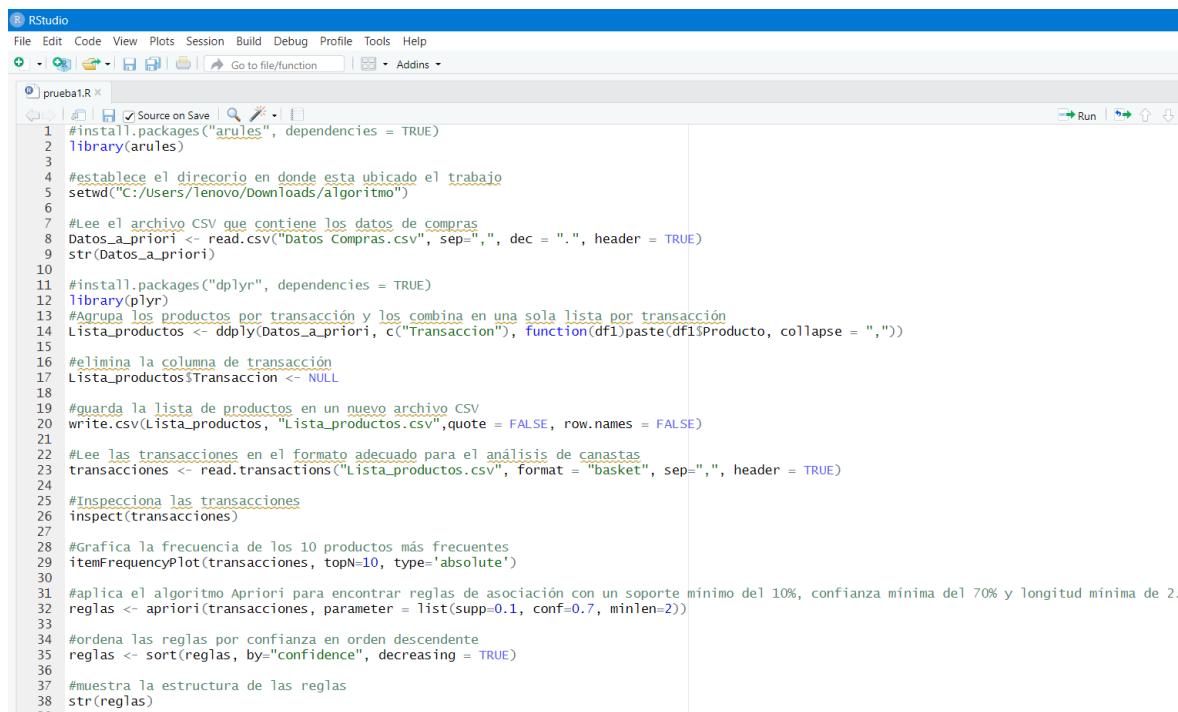
Comúnmente las empresas ofrecen múltiples tipos de productos o servicios y en una transacción de venta, dentro esto el algoritmo Apriori determina Patrones de venta dentro de las transacciones de una empresa y con esto podemos ayudar a definir las promociones o estrategias de venta que maximicen una mayor productividad.

El resultado del algoritmo a priori son las reglas de asociación que nos indican los patrones de ocurrencia que hay dentro de un conjunto de colecciones.

Problemática.

En una tienda se registraron 12 transacciones que incluyen diversos productos similares. Estos productos presentan múltiples asociaciones entre sí, y se desea identificar las reglas de asociación más frecuentes para entender mejor estas relaciones. Para ello, se emplearon métricas como el soporte, la confianza y el lift, utilizando un programa que implementa el algoritmo Apriori para su análisis.

Para la solución del problema se utilizo el programa RStudio en donde se realizó el código, como se muestra a continuación:



```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
prueba1.R x
Source on Save Run
1 #install.packages("arules", dependencies = TRUE)
2 library(arules)
3
4 #establece el directorio en donde esta ubicado el trabajo
5 setwd("C:/Users/Ilenovo/Downloads/álgortimo")
6
7 #Lee el archivo CSV que contiene los datos de compras
8 Datos_a_priori <- read.csv("Datos Compras.csv", sep=",", dec = ".", header = TRUE)
9 str(Datos_a_priori)
10
11 #install.packages("dplyr", dependencies = TRUE)
12 library(dplyr)
13 #Agrupa los productos por transacción y los combina en una sola lista por transacción
14 Lista_productos <- dplyr::group_by(Datos_a_priori, Transaccion) %>% paste(df1$Producto, collapse = ",") %>% unlist()
15
16 #elimina la columna de transacción
17 Lista_productos$Transaccion <- NULL
18
19 #guarda la lista de productos en un nuevo archivo CSV
20 write.csv(Lista_productos, "Lista_productos.csv", quote = FALSE, row.names = FALSE)
21
22 #Lee las transacciones en el formato adecuado para el análisis de canastas
23 transacciones <- read.transactions("Lista_productos.csv", format = "basket", sep = ",", header = TRUE)
24
25 #Inspecciona las transacciones
26 inspect(transacciones)
27
28 #Grafica la frecuencia de los 10 productos más frecuentes
29 itemFrequencyPlot(transacciones, topN=10, type='absolute')
30
31 #aplica el algoritmo Apriori para encontrar reglas de asociación con un soporte mínimo del 10%, confianza mínima del 70% y longitud mínima de 2.
32 reglas <- apriori(transacciones, parameter = list(supp=0.1, conf=0.7, minlen=2))
33
34 #ordena las reglas por confianza en orden descendente
35 reglas <- sort(reglas, by="confidence", decreasing = TRUE)
36
37 #muestra la estructura de las reglas
38 str(reglas)
```

En esta parte de aquí se especifica y se agrega el archivo en donde se tienen todos los datos de las transacciones, así mismo lee el archivo de las transacciones, y puede graficar en este caso los productos mas frecuentes. Por otra parte, se agrega el algoritmo A priori con los parámetros de soporte y confianza, después ordena las reglas por confianza en orden descendente y después muestra las reglas.

```
#identifica reglas redundantes
redundantes <- is.redundant(reglas)
redundantes

#muestra los índices de las reglas redundantes
which(is.redundant(reglas))

#elimina las reglas redundantes
reglas_podadas <- reglas[!redundantes]

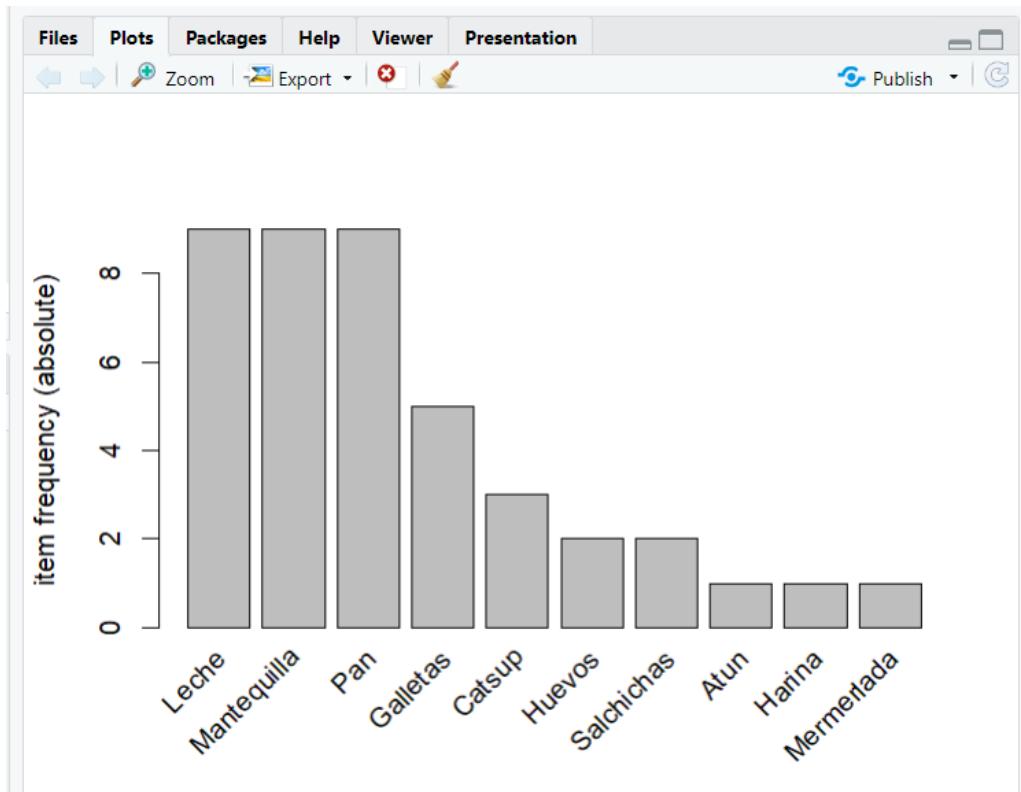
#inspecciona las reglas podadas
inspect(reglas_podadas)

#install.packages("arulesViz", dependencies = TRUE)
library(arulesViz) #carga el paquete arulesViz para visualización de reglas

#grafica las reglas de asociación usando un gráfico interactivo
plot(reglas_podadas, method="graph", engine="interactive", shading="confidence")
```

En esta parte de aquí, se identifican las reglas redundantes quiere decir las que se lleguen a repetir, así mismo, se eliminan las reglas redundantes. Después inspecciona las reglas podadas (las reglas más frecuentes, ósea los conjuntos mas frecuentes) y las grafica.

Después se muestra la gráfica, de los productos más frecuentes:



```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to function Addins
prueba1.R
Source on Save Run Source Environment History Connections Tutorial
1 #install.packages("arules", dependencies = TRUE)
2 library(arules)
3
4 #establece el directorio en donde esta ubicado el trabajo
5 setwd("C:/Users/lenovo/Downloads/algortimo")
6
7 #Lee el archivo CSV que contiene los datos de compras
8 Datos_a_priori <- read.csv("Datos Compras.csv", sep = ";", dec = ".", header = TRUE)
9 str(Datos_a_priori)
10
11 #install.packages("dplyr", dependencies = TRUE)
12 library(dplyr)
13 #Agrupa los productos por transacción y los combina en una sola lista por transacción
14 Lista_productos <- dplyr::group_by(Datos_a_priori, c("Transaccion"), .function=df1$paste(df1$Producto, collapse = ""))
15
16 #elimina la columna de transacción
17 Lista_productos$Transaccion <- NULL
18
19 #guarda la lista de productos en un nuevo archivo CSV
20 write.csv(Lista_productos, "Lista_productos.csv", quote = FALSE, row.names = FALSE)
21
22 #Lee las transacciones en el formato adecuado para el análisis de canastas
23 transacciones <- read.transactions("Lista_productos.csv", format = "basket", sep = ";", header = TRUE)
24
25 #Inspecciona las transacciones
26 inspect(transacciones)
27
28
10.1 (Top Level) :

```

R Script

Console Terminal

R 4.4.1 - C:/Users/lenovo/Downloads/algortimo/

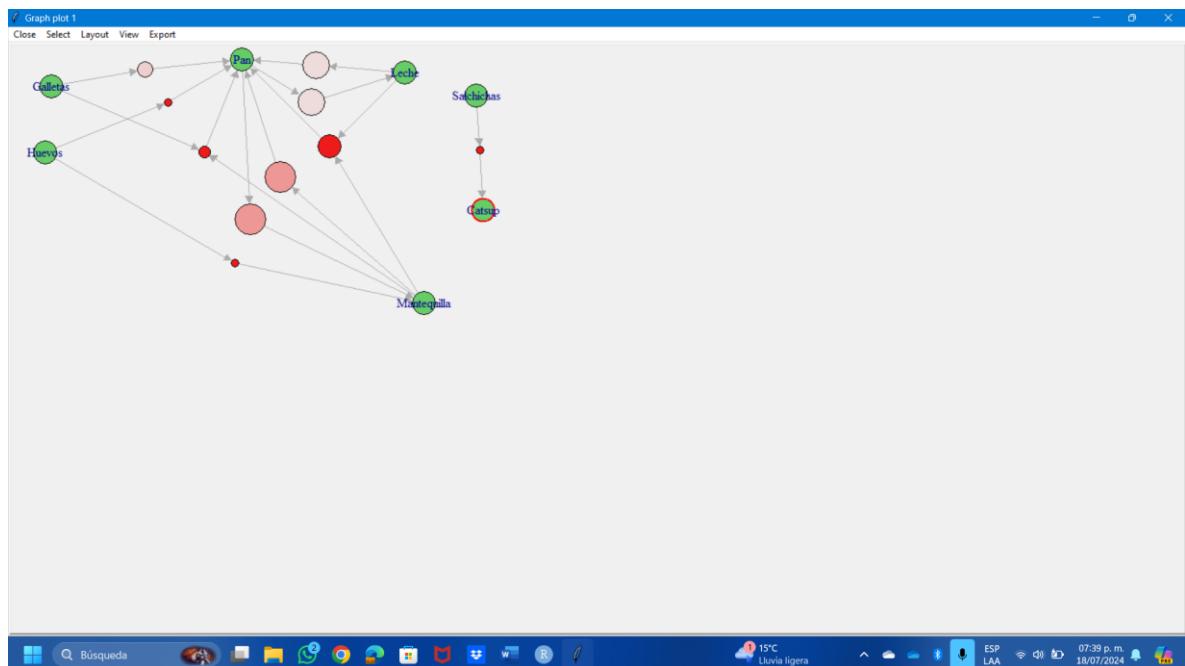
```

[1] {Huevos}          => {Mantequilla} 0.1666667 1.0000000 0.1666667 1.333333 2
[2] {Huevos}          => {Pan}         0.1666667 1.0000000 0.1666667 1.333333 2
[3] {Salchichas}      => {Catsup}     0.1666667 1.0000000 0.1666667 4.000000 2
[4] {Galletas, Mantequilla} => {Pan}        0.2500000 1.0000000 0.2500000 1.333333 3
[5] {Leche, Mantequilla} => {Pan}        0.5000000 1.0000000 0.5000000 1.333333 6
[6] {Mantequilla}     => {Pan}         0.4000000 0.8888889 0.7500000 1.185185 8
[7] {Pan}              => {Mantequilla} 0.6666667 0.8888889 0.7500000 1.037037 7
[8] {Galletas}        => {Pan}         0.3333333 0.8000000 0.4166667 4
[9] {Leche}            => {Pan}         0.5833333 0.7777778 0.7500000 1.037037 7
[10] {Pan}             => {Leche}       0.5833333 0.7777778 0.7500000 1.037037 7

```

15°C Lluvia 07:24 p.m. 18/07/2024

En la ejecución, se puede observar las transacciones frecuentes en donde se muestra el soporte, la confianza y el lift de cada transacción. Así podemos determinar las reglas de asociación de cada compra.



Así se muestra el algoritmo a priori, sobre las transacciones.

Conclusión.

El algoritmo Apriori es una herramienta poderosa para descubrir patrones y relaciones interesantes en grandes volúmenes de datos transaccionales, proporcionando información valiosa para la toma de decisiones en diversos campos de aplicación, proporcionando datos valiosos para optimizar procesos empresariales y mejorar la eficiencia operativa. El uso del algoritmo Apriori va más allá de las simples cestas de mercado, extendiéndose a diversos sectores donde el análisis de datos puede proporcionar ventajas competitivas significativas. La capacidad de interpretar y actuar sobre los patrones descubiertos permite a las organizaciones ser más proactivas y eficientes, asegurando su relevancia y éxito en un mercado cada vez más competitivo.



GOBIERNO DEL
ESTADO DE
MÉXICO



ASOCIACIÓN A LA MINERÍA DE DATOS

RAKESH AGRAWAL



UNIVERSIDAD POLITÉCNICA DEL VALLE DE
TOLUCA.

INGENIERÍA EN TECNOLOGÍAS DE LA
INFORMACIÓN.

FACILITADOR: M.A. JOSÉ LUIS LEDEZMA SÁNCHEZ.

ALUMNO: COLIN VIEYRA JONATHAN IVAN

ALUMNO: MENDOZA GARCIA EMMANUEL

ITI9MA.



CONTENIDO



Introducción

Objetivo

Algoritmo Apriori

Definiciones

Conjuntos de elementos
(Itemset)

Tipos de conjuntos de
elementos

Frecuencia de conjuntos
de elementos

Representación de una
base de datos
transaccional

¿Que son las reglas de
asociación?

Soporte

Confianza

Lift (Levantamiento)

Ventajas y desventajas

Ejemplo

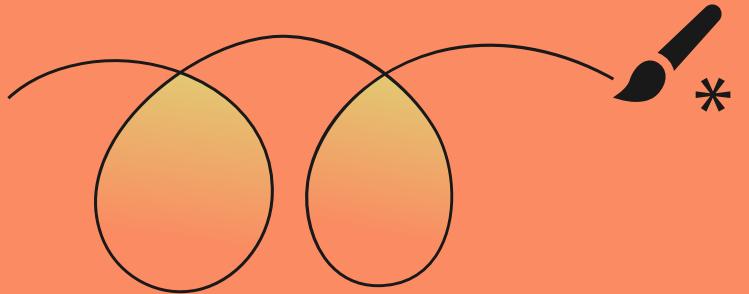
Conclusión

INTRODUCCIÓN

La minería de datos se ha considerado una alternativa para dar solución a diferentes problemáticas que requieren el análisis de grandes volúmenes de datos.

En esta presentación se abarcara el tema de las asociaciones Agrawal (Algoritmo Apriori). Las reglas de asociación fueron utilizadas para el análisis de las cestas de mercado, a partir de dicho estudio se logró establecer criterios para tomar decisiones en base al comportamiento de compra de los productos por parte de los clientes.

OBJETIVO



Objetivo: encontrar asociaciones o correlaciones entre los elementos u objetos de bases de datos transaccionales, relacionales o data-warehouses.



- Rakesh Agrawal es un científico informático que ha hecho numerosas contribuciones a la minería de datos y a las bases de datos.
- Ramakrishnan Srikant ha colaborado en diversos trabajos relacionados con la minería de datos y la privacidad de datos.



El algoritmo Apriori

El algoritmo Apriori fue propuesto por Rakesh Agrawal y Ramakrishnan Srikant en 1994.

El desarrollo del algoritmo Apriori fue motivado por la necesidad de encontrar patrones de asociación en grandes bases de datos transaccionales, especialmente en el ámbito del comercio



Comúnmente las empresas ofrecen múltiples tipos de productos o servicios y en una transacción de venta, dentro esto el algoritmo Apriori determina Patrones de venta dentro de las transacciones de una empresa y con esto podemos ayudar a definir las promociones o estrategias de venta que maximicen una mayor productividad.

El resultado del algoritmo a priori son las reglas de asociación que nos indican los patrones de ocurrencia que hay dentro de un conjunto de colecciones.

Por ejemplo, dentro de un dominio de supermercado se podría tener la siguiente regla de asociación: que aplica un si -> entonces (si yo compro cereal, entonces compro leche)



DEFINICIONES

Elemento Item:

Un objeto o producto individual.

Conjunto de elementos (Itemset):

Un conjunto de uno o más elementos.

Elementos u objetos (items): Dependiendo del campo de aplicación, pueden ser productos, objetos, pacientes, eventos.

Transacción (transaction): Identificada con un identificador único, es un conjunto de elementos con un mínimo de un elemento. Los elementos pueden pertenecer a varias transacciones.

Base de datos transaccional: conjunto de transacciones. Cada transacción viene representada por su conjunto de items

Conjunto de elementos (itemset): Un grupo de elementos. Se pueden encontrar en una o varias transacciones.



APLICACIONES

Cesta de la compra

Asociación entre artículos comprados.
Por ejemplo, una regla de tipo {huevo} ⇒ {leche} implica que es muy frecuente la compra de pan y leche juntos



Biblioteca

Los libros que compra un cliente en una librería.



Páginas Web

Las páginas web visitadas por un usuario.

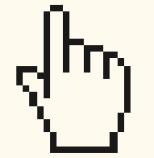


CONJUNTO DE ELEMENTOS (ITEMSET)

Un conjunto de elementos (o itemset) es un conjunto de uno o más elementos que se consideran juntos en una transacción o registro de una base de datos. En el contexto de la minería de datos y, específicamente, en la generación de reglas de asociación, los conjuntos de elementos son fundamentales para identificar patrones de co-ocurrencia en los datos.



TIPOS DE CONJUNTOS DE ELEMENTOS



Conjunto de 1 elemento (1-itemset):

Son conjunto que que contienen solo un elemento.

Como por ejemplo: {Leche}, {Pan}, {Mantequilla}.

Conjuntos de 2-Elementos (2-itemset):

Son conjuntos que contienen dos elementos.

Como po ejemplo: {Leche, Pan}, {Leche, Mantequilla}, {Pan, Mantequilla}.

Conjuntos de k-Elementos (k-itemset):

Son conjuntos que contienen k elementos.

Como po ejemplo: {Leche, Pan, Mantequilla} es un 3-itemset.

FRECUENCIA DE CONJUNTOS DE ELEMENTOS

Conjunto de elementos frecuentes

Se considera frecuente si su soporte (la proporción de transacciones en la base de datos que contiene el conjunto de elementos) es mayor o igual a un umbral mínimo de soporte predefinido. Como por ejemplo: Que sea mayor a un 50%

Conjunto de elementos infrecuentes

Se considera infrecuente si su soporte es menor que el umbral mínimo de soporte. Como por ejemplo: Que sea menor del 50%

FORMAS EN LAS QUE PODEMOS VER REPRESENTADA UNA BASE DE DATOS TRANSACCIONAL

TID	items
1	{leche, cerveza}
2	{leche, pan, huevos}
3	{pan, servilletas}
4	{leche, pan, huevos, servilletas}

leche	cerveza	pan	huevos	servilletas
1	1	1	1	3
2		2	3	5
3			4	
4			5	

TID	leche	cerveza	pan	huevos	servilletas
1	1	1	0	0	0
2	1	0	1	1	0
3	0	0	1	0	1
4	1	0	1	1	1

Disposición horizontal o tipo basket, donde se representan un conjunto de artículos por fila:

Disposición vertical, con artículo por columna:

Representación binaria (1 = artículo presente, 0 = artículo ausente):



EL ALGORITMO

Definición del Algoritmo

Inicio:

Sea C_1 el conjunto formado por todos los items de la base.

Sea $L_1 \subseteq C_1$ los items que son frecuentes, es decir $L_1 = \{i_k : supp(i_k) \geq U\}$.

Repite para $k = 2, 3, \dots$ (determinación de conjuntos de k elementos):

1. Paso C_k : se proponen todos los posibles conjuntos candidatos de k elementos que surjen de combinar los elementos de L_{k-1} .
2. Paso L_k : se filtran (eliminan) de C_k los candidatos no frecuentes de L_k .

Fin: El algoritmo se detiene cuando todos los conjuntos candidatos que se proponen no superan el umbral U .

EJEMPLO DE CÓMO FUNCIONA EL ALGORITMO

Tenemos dentro de esta tabla siguiente n=5 transacciones con p=4 ítems. Se tiene un umbral de u=2 y apliquemos el algoritmo definido.

Transacciones	Ítems	C1	#	L1	#
1	A,B,C	A	3	A	3
2	A	B	3	B	3
3	B	C	3	C	3
4	A,C	D	1		
5	B,C,D				

U=2

EJEMPLO DE CÓMO FUNCIONA EL ALGORITMO

EJEMPLO DE CÓMO FUNCIONA EL ALGORITMO

Transacciones	Ítems	C1	#	L1	#	C2	#	L2	#	C3	#	L3	#
1	A,B,C	A	3	A	3	AB	1	AC	2	ABC	1	ABC	0
2	A	B	3	B	3	AC	2	BC	2				
3	B	C	3	C	3	BC	2						
4	A,C	D	1										
5	B,C,D												

U=2

Este ejemplo, nos asegura haber recolectado todos los subconjuntos de uno o mas elementos que satisfacen el umbral=2

REGLAS DE ASOCIACIÓN



Fueron desarrolladas por Rakesh Agrawal y Ramakrishnan Srikant, son una técnica muy utilizada en la minería de datos que busca descubrir relaciones interesantes y útiles entre conjuntos de elementos en grandes bases de datos.



Estas reglas son especialmente útiles en el análisis de transacciones de ventas, donde se desea identificar patrones de compra comunes..

CONCEPTOS DE LAS REGLAS DE ASOCIACIÓN AGRAWAL

Soporte (Support): El soporte de un itemset (conjunto de elementos) se refiere a la proporción de transacciones en la base de datos que contienen ese itemset. Es una medida de la frecuencia con la que un conjunto específico de items aparece en las transacciones.

La formula para calcular el soporte es la siguiente:

$$Soporte(X) = \frac{\text{Num. de transacciones con } X}{n}$$

Conjunto de Elementos

Número de Transacciones

Confianza (Confidence): Consiste en la probabilidad de que tanto aparece el consecuente en las transacciones que contiene el antecedente.

A → Antecedente

B → Consecuente

La formula para calcular la confianza es la siguiente:

$$\text{Confianza}(A \rightarrow C) = \frac{\text{Conteo}(\text{Antecedente} \cup \text{Consecuente})}{\text{Conteo}(\text{Antecedente})}$$

Lift (Levantamiento): Es el soporte (support) del conjunto de elementos que agrupa A y B, dividido por el soporte de A y el soporte de B. Este valor puede ser cualquier número real positivo.

A → Antecedente
B → Consecuente

La formula para calcular el lift es la siguiente:

$$\text{Lift}(A \rightarrow C) = \frac{\text{Soporte}(Antecedente \rightarrow Consecuente)}{\text{Soporte}(Antecedente) * \text{Soporte}(Consecuente)}$$

- un lift >1: indica un efecto positivo de A en B (o B en A).
- un lift =1: significa que no hay efecto, y es como si los elementos o conjuntos de elementos fueran independientes.
- un lift <1: significa que hay un efecto negativo de A en B o viceversa, como si fueran excluyentes entre sí.

EJEMPLO:

En un supermercado se realizaron las siguientes transacciones de diferentes clientes, a continuación, se muestran las compras que se realizaron de cada uno:

Compra #1



Compra #2



Compra #3



SOLUCIÓN:



Identificar los conjuntos de elementos de cada compra:

Compra #1

{LECHE, HUEVO}
{LECHE, PAN}
{PAN, HUEVO}
{LECHE, PAN, HUEVO}

Compra #2

{CEREAL, LECHE}
{CEREAL, CREMA}
{CEREAL, HUEVO}
{LECHE, CREMA}
{LECHE, HUEVO}
{CREMA, HUEVO}
{CEREAL, LECHE, CREMA}
{CEREAL, LECHE, HUEVO}
{CEREAL, CREMA, HUEVO}
{LECHE, CREMA, HUEVO}

Compra #3

{LECHE, CREMA}
{LECHE, PAN}
{LECHE, HUEVO}
{CREMA, PAN}
{CREMA, HUEVO}
{PAN, HUEVO}
{LECHE, CREMA, PAN}
{LECHE, CREMA, HUEVO}
{LECHE, PAN, HUEVO}
{CREMA, PAN, HUEVO}

SOLUCIÓN:

Calculamos el soporte de cada compra:

Conjunto de Elementos

$$\text{Soporte}(X) = \frac{\text{Num. de transacciones con } X}{n}$$

Número de Transacciones

Compra #1

{LECHE, HUEVO}
{LECHE, PAN}
{PAN, HUEVO}
{LECHE, PAN, HUEVO}

Compra #2

{CEREAL, LECHE}
{CEREAL, CREMA}
{CEREAL, HUEVO}
{LECHE, CREMA}
{LECHE, HUEVO}
{CREMA, HUEVO}
{CEREAL, LECHE, CREMA}
{CEREAL, LECHE, HUEVO}
{CEREAL, CREMA, HUEVO}
{LECHE, CREMA, HUEVO}

Compra #3

{LECHE, CREMA}
{LECHE, PAN}
{LECHE, HUEVO}
{CREMA, PAN}
{CREMA, HUEVO}
{PAN, HUEVO}
{LECHE, CREMA, PAN}
{LECHE, CREMA, HUEVO}
{LECHE, PAN, HUEVO}
{CREMA, PAN, HUEVO}

Soporte de los Conjuntos

{LECHE, HUEVO}	= 3/3 = 1.00
{LECHE, PAN}	= 2/3 = 0.66
{PAN, HUEVO}	= 2/3 = 0.66
{LECHE, PAN, HUEVO}	= 2/3 = 0.66
{LECHE, CREMA}	= 2/3 = 0.66
{CREMA, HUEVO}	= 2/3 = 0.66
{LECHE, CREMA, HUEVO}	= 2/3 = 0.66
Todos los demás	= 1/3 = 0.33

SOLUCIÓN:

Creación de reglas de asociación:

Se crean reglas apartir
de los conjuntos de
itemset más frecuentes.

De: $>0.66 = 66\%$

- {LECHE, PAN}
- {LECHE, HUEVO}
- {PAN, HUEVO}
- {LECHE, CREMA}
- {CREMA, HUEVO}
- {LECHE, PAN, HUEVO}
- {LECHE, CREMA, HUEVO}



SOLUCIÓN:

Conjuntos
Frecuentes

REGLAS

{LECHE, PAN}

$\{LECHE\} \rightarrow \{PAN\}$
 $\{PAN\} \rightarrow \{LECHE\}$

{LECHE, HUEVO}

$\{LECHE\} \rightarrow \{HUEVO\}$
 $\{HUEVO\} \rightarrow \{LECHE\}$

{PAN, HUEVO}

$\{PAN\} \rightarrow \{HUEVO\}$
 $\{HUEVO\} \rightarrow \{PAN\}$

{LECHE, CREMA}

$\{LECHE\} \rightarrow \{CREMA\}$
 $\{CREMA\} \rightarrow \{LECHE\}$

Conjuntos
Frecuentes

REGLAS

{CREMA, HUEVO}

$\{CREMA\} \rightarrow \{HUEVO\}$
 $\{HUEVO\} \rightarrow \{CREMA\}$

{LECHE, PAN, HUEVO}

$\{LECHE, PAN\} \rightarrow \{HUEVO\}$
 $\{LECHE, HUEVO\} \rightarrow \{PAN\}$
 $\{PAN, HUEVO\} \rightarrow \{LECHE\}$

{LECHE, CREMA, HUEVO}

$\{LECHE, CREMA\} \rightarrow \{HUEVO\}$
 $\{LECHE, HUEVO\} \rightarrow \{CREMA\}$
 $\{CREMA, HUEVO\} \rightarrow \{LECHE\}$

SOLUCIÓN:

Calcular el soporte de las reglas frecuentes:

$$\text{Soporte}(A \rightarrow C) = \frac{\text{Conteo}(A \cup C)}{n}$$

Número de Transacciones

SOPORTE DE LAS REGLAS

$$\{\text{LECHE}\} \rightarrow \{\text{PAN}\} \quad \frac{\text{Conteo}(\{\text{LECHE}\} \cup \{\text{PAN}\})}{n} = \frac{2}{3} = \mathbf{0.6\bar{6}}$$

$$\{\text{PAN}\} \rightarrow \{\text{LECHE}\} \quad \frac{\text{Conteo}(\{\text{PAN}\} \cup \{\text{LECHE}\})}{n} = \frac{2}{3} = \mathbf{0.6\bar{6}}$$

$$\{\text{LECHE}, \text{CREMA}\} \rightarrow \{\text{HUEVO}\} \quad \frac{\text{Conteo}(\{\text{LECHE}, \text{CREMA}\} \cup \{\text{HUEVO}\})}{n} = \frac{2}{3} = \mathbf{0.6\bar{6}}$$

$$\{\text{LECHE}, \text{HUEVO}\} \rightarrow \{\text{CREMA}\} \quad \frac{\text{Conteo}(\{\text{LECHE}, \text{HUEVO}\} \cup \{\text{CREMA}\})}{n} = \frac{2}{3} = \mathbf{0.6\bar{6}}$$



Compra #1



Compra #2



Compra #3

SOLUCIÓN:

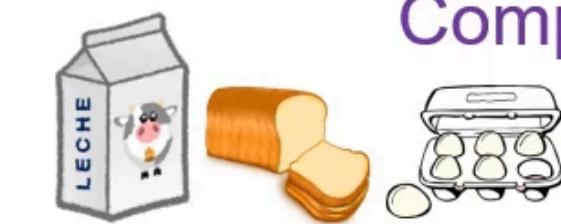
Calcular la confianza de las reglas frecuentes:

$$\text{Confianza}(A \rightarrow C) = \frac{\text{Conteo}(Antecedente \cup Consecuente)}{\text{Conteo}(Antecedente)}$$

CONFIANZA DE LAS REGLAS

{LECHE} → {PAN}

$$\frac{\text{Conteo}(\{LECHE\} \cup \{PAN\})}{\text{Conteo}(\{LECHE\})} = \frac{2}{3} = 0.66$$



Compra #1

{PAN} → {LECHE}

$$\frac{\text{Conteo}(\{PAN\} \cup \{LECHE\})}{\text{Conteo}(\{PAN\})} = \frac{2}{2} = 1.00$$



Compra #2

{LECHE, CREMA} → {HUEVO}

$$\frac{\text{Conteo}(\{LECHE, CREMA\} \cup \{HUEVO\})}{\text{Conteo}(\{LECHE, CREMA\})} = \frac{2}{2} = 1.00$$



Compra #3

{LECHE, HUEVO} → {CREMA}

$$\frac{\text{Conteo}(\{LECHE, HUEVO\} \cup \{CREMA\})}{\text{Conteo}(\{LECHE, HUEVO\})} = \frac{2}{3} = 0.66$$

SOLUCIÓN:

Calcular el lift de las reglas frecuentes:

$$\text{Lift}(A \rightarrow C) = \frac{\text{Soporte(Antecedente} \rightarrow \text{Consecuente})}{\text{Soporte(Antecedente)} * \text{Soporte(Consecuente)}}$$

LEVANTAMIENTO DE LAS REGLAS

$$\{\text{LECHE}\} \rightarrow \{\text{PAN}\} \quad \frac{\text{Soporte}(\{\text{LECHE}\} \rightarrow \{\text{PAN}\})}{\text{Soporte}(\{\text{LECHE}\}) * \text{Soporte}(\{\text{PAN}\})} = \frac{0.66}{(3/3) * (2/3)} = 1$$



$$\{\text{PAN}\} \rightarrow \{\text{LECHE}\} \quad \frac{\text{Soporte}(\{\text{PAN}\} \rightarrow \{\text{LECHE}\})}{\text{Soporte}(\{\text{PAN}\}) * \text{Soporte}(\{\text{LECHE}\})} = \frac{0.66}{(2/3) * (3/3)} = 1$$



$$\{\text{LECHE, CREMA}\} \rightarrow \{\text{HUEVO}\} \quad \frac{\text{Soporte}(\{\text{LECHE, CREMA}\} \rightarrow \{\text{HUEVO}\})}{\text{Soporte}(\{\text{LECHE, CREMA}\}) * \text{Soporte}(\{\text{HUEVO}\})} = \frac{0.66}{(2/3) * (3/3)} = 1$$

$$\{\text{LECHE, HUEVO}\} \rightarrow \{\text{CREMA}\} \quad \frac{\text{Soporte}(\{\text{LECHE, HUEVO}\} \rightarrow \{\text{CREMA}\})}{\text{Soporte}(\{\text{LECHE, HUEVO}\}) * \text{Soporte}(\{\text{CREMA}\})} = \frac{0.66}{(3/3) * (2/3)} = 1$$

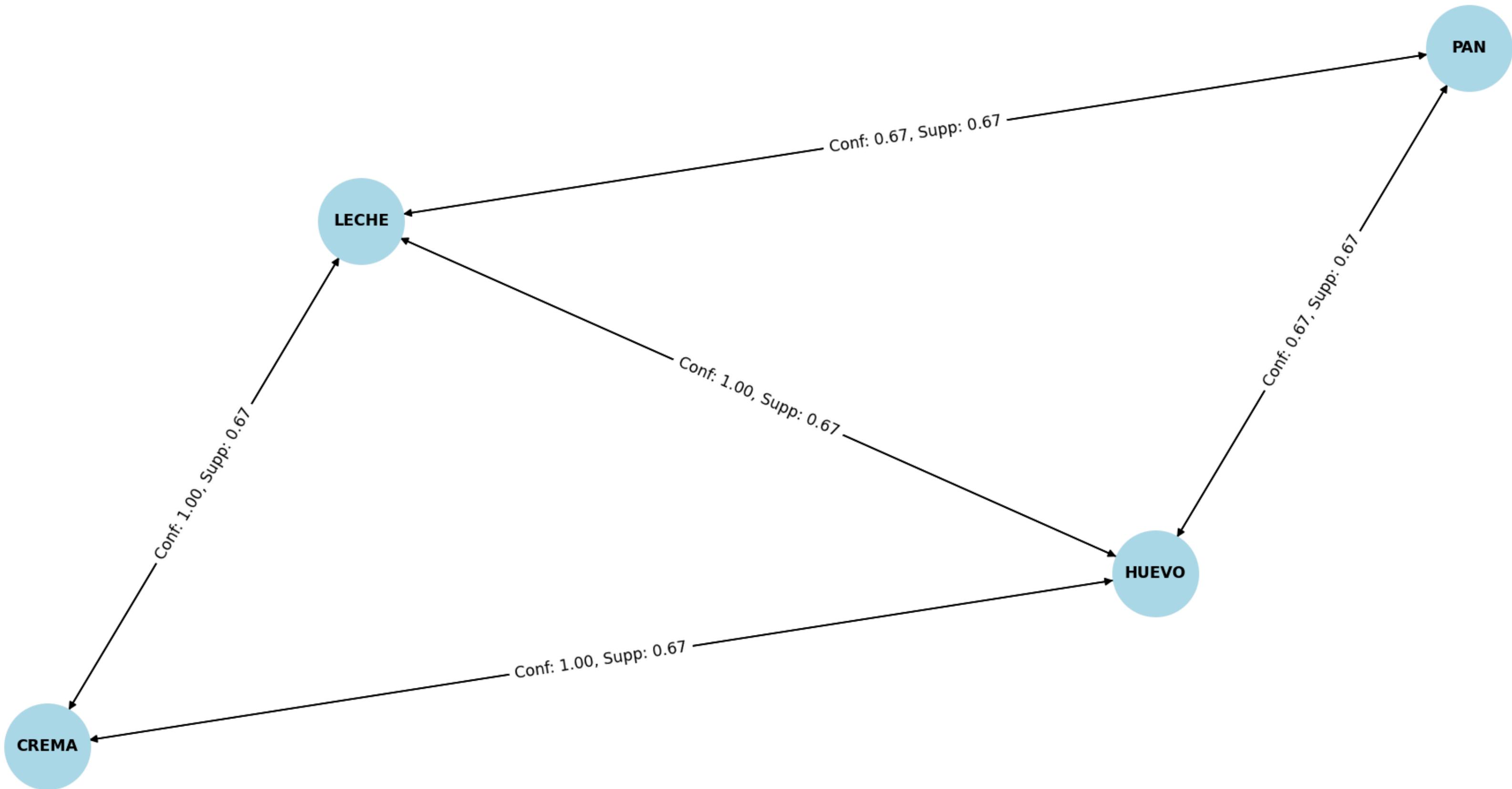


```
• Ejemplo1.py > ...
1  from efficient_apriori import apriori
2  import networkx as nx
3  import matplotlib.pyplot as plt
4
5  # Datos de transacciones
6  transacciones = [("LECHE", "PAN", "HUEVO"),
7  |   |   |   ("CEREAL", "LECHE", "CREMA", "HUEVO"),
8  |   |   |   ("LECHE", "CREMA", "PAN", "HUEVO")]
9
10 # Generando conjuntos frecuentes y reglas de asociación
11 conjuntos, reglas = apriori(transacciones,
12 |   |   |   |   min_support = 0.6,
13 |   |   |   |   min_confidence = 0.6)
14
15 # Filtrando reglas con consecuente de un elemento
16 reglas = list(filter(lambda regla: len(regla.rhs) == 1, reglas))
17
18 # Imprimiendo reglas
19 for regla in reglas:
20     print(regla)
21
22 # Creando el grafo
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\lenovo\Downloads\Python> & "C:/Program Files/Python/python.exe" c:/users/lenovo/Downloads/Python/Ejemplo1.py
{HUEVO} -> {CREMA} (conf: 0.667, supp: 0.667, lift: 1.000, conv: 1.000)
{CREMA} -> {HUEVO} (conf: 1.000, supp: 0.667, lift: 1.000, conv: 0.000)
{LECHE} -> {CREMA} (conf: 0.667, supp: 0.667, lift: 1.000, conv: 1.000)
{CREMA} -> {LECHE} (conf: 1.000, supp: 0.667, lift: 1.000, conv: 0.000)
{LECHE} -> {HUEVO} (conf: 1.000, supp: 1.000, lift: 1.000, conv: 0.000)
{HUEVO} -> {LECHE} (conf: 1.000, supp: 1.000, lift: 1.000, conv: 0.000)
{PAN} -> {HUEVO} (conf: 1.000, supp: 0.667, lift: 1.000, conv: 0.000)
{HUEVO} -> {PAN} (conf: 0.667, supp: 0.667, lift: 1.000, conv: 1.000)
{PAN} -> {LECHE} (conf: 1.000, supp: 0.667, lift: 1.000, conv: 0.000)
{LECHE} -> {PAN} (conf: 0.667, supp: 0.667, lift: 1.000, conv: 1.000)
{HUEVO, LECHE} -> {CREMA} (conf: 0.667, supp: 0.667, lift: 1.000, conv: 1.000)
{CREMA, LECHE} -> {HUEVO} (conf: 1.000, supp: 0.667, lift: 1.000, conv: 0.000)
```

Figure 1



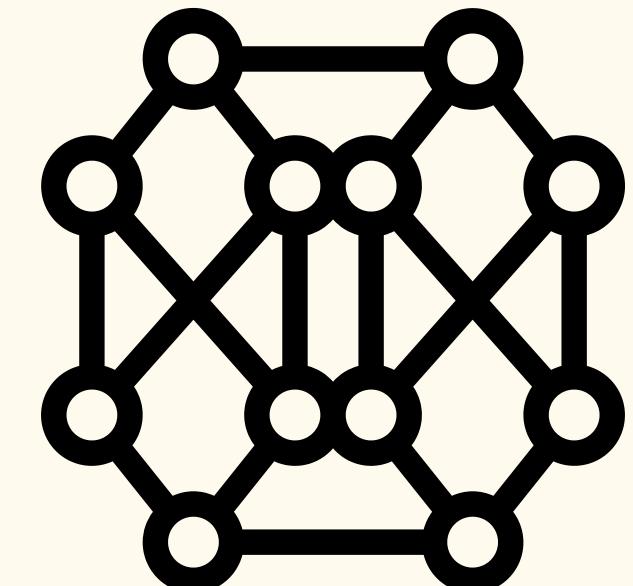
VENTAJAS

1. Es fácil de entender e implementar

2. Funciona bien con bases de datos pequeñas y medianas.

3. Identifica de manera efectiva conjuntos de elementos frecuentes a partir de los cuales se pueden generar reglas de asociación.

4. Las reglas generadas son fácilmente interpretables, lo que facilita la comprensión de las relaciones entre los datos.



DESVENTAJAS

1. El algoritmo puede ser muy lento y consumir mucha memoria, especialmente cuando se trabaja con bases de datos grandes.
2. No es adecuado para bases de datos muy grandes debido a su alta complejidad computacional.
3. Puede producir una gran cantidad de reglas, muchas de las cuales pueden ser triviales o irrelevantes, dificultando la identificación de reglas verdaderamente útiles.
4. Si los datos contienen muchos elementos infrecuentes, puede ser difícil encontrar reglas significativas.

CONCLUSIÓN

el algoritmo Apriori es una herramienta poderosa para descubrir patrones y relaciones interesantes en grandes volúmenes de datos transaccionales, proporcionando información valiosa para la toma de decisiones en diversos campos de aplicación.

REFERENCIAS:

RPubs - Reglas de asociación. (2020, February 27). Rpubs.com.
https://rpubs.com/Cristina_Gil/Reglas_Asociacion

Joaquín Amat Rodrigo. (2015). Reglas de asociación y algoritmo Apriori con R. Cienciadedatos.net. https://cienciadedatos.net/documentos/43_reglas_de_asociacion