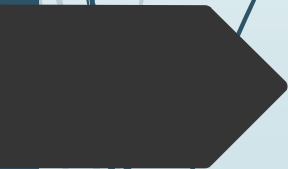


REGLAS DE ASOCIACION



BASE DE DATOS II – LSI- 2019

REGLAS DE ASOCIACION



Lic. Nevelin Irene Salazar

Reglas de Asociación

- Permiten expresar relaciones entre items de una BD. Son aplicables a la toma de decisiones.
- Ejemplos
 - Relación en la compra de productos
 - Itinerarios más utilizados por los visitantes de páginas WEB

Reglas de Asociación

■ Definición

- Sea I el conjunto de ítems de una base de datos D .
- Una *Regla de Asociación* (RA) es una implicación de la forma

$$X \Rightarrow Y$$

donde $X \subset I$, $Y \subset I$, y $X \cap Y = \emptyset$.

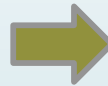
Análisis de la Canasta de Mercado

- Los productos comprados por los clientes de un supermercado se representan como transacciones $T_j = \{I_1, I_2, \dots, I_k\}$

ID	Productos (o ítems)
Juan	Papas, Huevos, Jamón
María	Pan, Leche, Huevos
Luis	Pan, Leche, Papas, Huevos
Ana	Pan, Leche

Reglas de Asociación. Ejemplo

ID	Productos
Juan	Papas, Huevos, Jamón
María	Pan, Leche, Huevos
Luis	Pan, Leche, Papas, Huevos
Ana	Pan, Leche



#	Regla
1	Pan \rightarrow Leche
2	Leche \rightarrow Pan
3	Papas \rightarrow Huevos
4	Huevos \rightarrow Papas
5	Pan y Huevos \rightarrow Leche
6	Leche \rightarrow Huevos y Pan
7	Pan \rightarrow Huevos y Leche

Problemas de las Reglas de Asociación.

- Aplicable únicamente a variables cualitativas o discretizadas.
- ¿Cómo limitamos el número de reglas? ¿Cómo hacemos manejable el proceso de procesamiento posterior?
- La respuesta esta en las **métricas** que usamos para medir la **importancia o interés de la regla**

Calidad de una regla

- Generalmente se usan tres medidas
 - **Soporte**
 - **Confianza o precisión**
 - **Lift**

SOPORTE

- Dada una regla, si $A \Rightarrow B$, el soporte de la regla se define como el **numero de veces que A y B aparecen juntos en una base de datos de transacciones**
- Soporte puede definirse para ítems individualmente o para una regla
- El soporte nos dice qué tan importante o interesante es un conjunto de elementos en función de su número de apariciones.
- El primer requisito que podemos imponer para limitar el numero de reglas, es que las reglas tengan un **soporte mínimo**.

Ejemplo Soporte

1000 trans

Jugo de
naranja
(400 trans)

Soporte(JN)=400

Soporte(JN)=400/1000=0.40

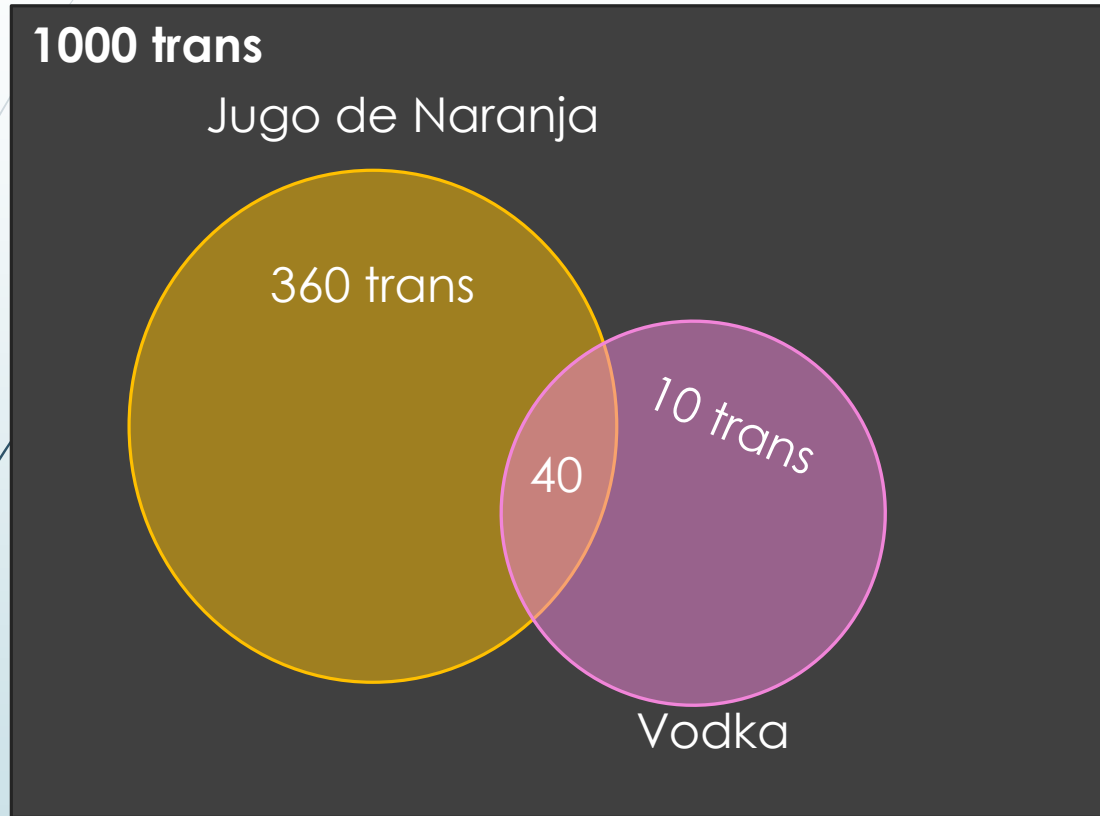
Soporte(V)=50

Soporte(V)=50/1000=0.05

1000 trans

Vodka
(50 trans)

Ejemplo Soporte



Soporte(JN y V)=40

Soporte(JN y V)=40/1000=0.04



Probabilidad
Conjunta

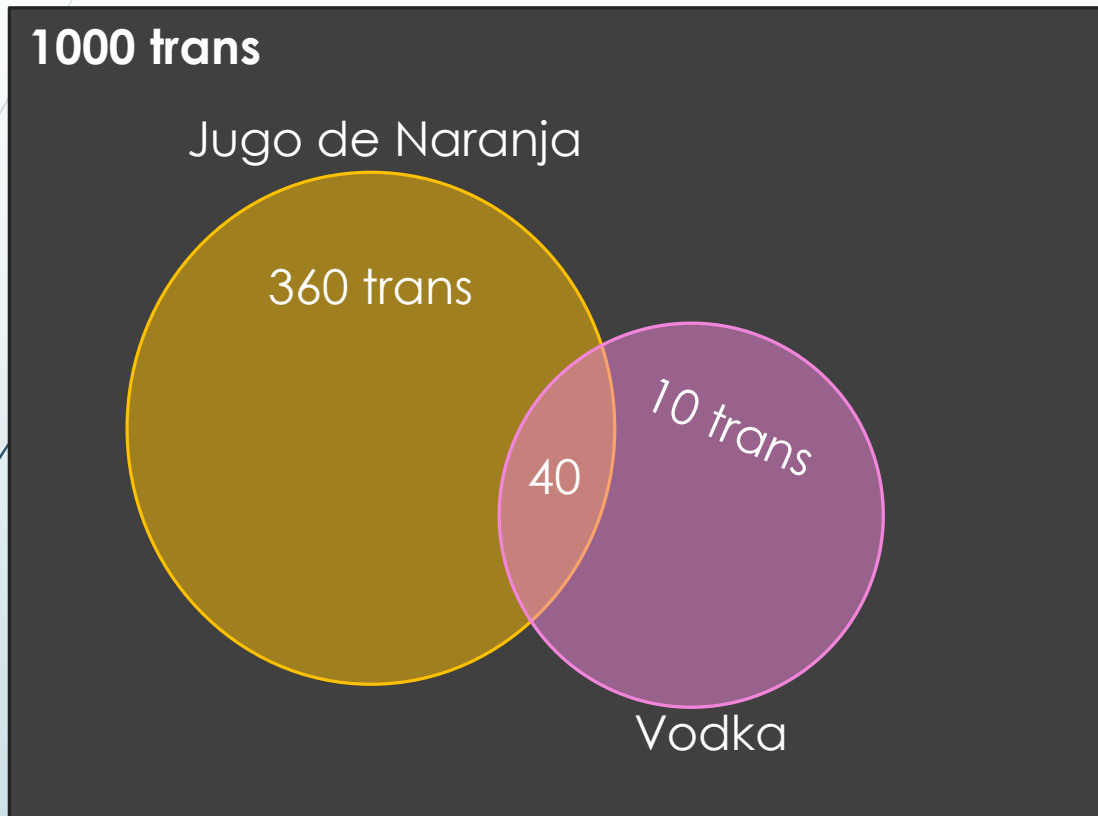
CONFIANZA

- Dada una regla, si $A \Rightarrow B$, la **confianza** de esta regla es el **cociente del soporte de la regla y el soporte del antecedente solamente**.
- **$\text{Confianza}(A \Rightarrow B) = S(A \Rightarrow B) / S(A)$**
- El **soporte** mide la frecuencia relativa, **confianza** mide la fortaleza de la regla.
- La confianza nos dice qué tan probable es un consecuente cuando ha ocurrido el antecedente

Cual es la Confianza de «Vodka \Rightarrow Jugo de Naranja»?

Cual es la Confianza de «Jugo de Naranja \Rightarrow Vodka»?

Ejemplo Confianza



$$\text{Confianza}(V \Rightarrow \text{JN}) = S(V \Rightarrow \text{JN}) / S(V) = \mathbf{40/50 = 0.80}$$

$$\text{Confianza}(\text{JN} \Rightarrow V) = S(\text{JN} \Rightarrow V) / S(\text{JN}) = \mathbf{40/400 = 0.10}$$

Dado el siguiente ejemplo calcular:

ítems	Compró Pan	No Compró Pan	Total
Compró JN	280	120	400
No Compró JN	420	180	600
Total	700	300	1000

- Soporte(Pan)=?
- Soporte(JN)=?
- Soporte(JN=>Pan) =
Soporte(Pan=>JN)=?
- Confianza(Pan=>JN)=?
- Confianza(JN=>Pan)=?

Solución del ejemplo:

ítems	Compró Pan	No Compró Pan	Total
Compró JN	280	120	400
No Compró JN	420	180	600
Total	700	300	1000

- **Soporte**(Pan)=**0.70**
- **Soporte**(JN)=**0.40**
- **Soporte**(JN=>Pan) =
Soporte(Pan=>JN)=**0.28**
- **Confianza**(Pan=>JN)= $0.28/0.70$ =**0.40**
- **Confianza**(JN=>Pan)= $0.28/0.40$ =**0.70**

Solución del ejemplo:

ítems	Compró Pan	No Compró Pan	Total
Compró JN	280	120	400
No Compró JN	420	180	600
Total	700	300	1000

- **Soporte**(Pan)=**0.70**
- **Soporte**(JN)=**0.40**
- **Soporte**(JN=>Pan) =
Soporte(Pan=>JN)=**0.28**
- **Confianza**(Pan=>JN)= $0.28/0.70$ =**0.40**
- **Confianza**(JN=>Pan)= $0.28/0.40$ =**0.70**

LIFT

- Nos dice qué tan probable es el consecuente cuando el antecedente ya ha ocurrido, teniendo en cuenta el soporte de ambos antecedentes y consecuentes

$$\text{Lift}(A \Rightarrow B) = \text{Soporte}(A \Rightarrow B) / \{\text{Soporte}(A) * \text{Soporte}(B)\}$$

- Cuantifica la relación existente entre A y B
- **Lift = 1** o muy cerca a 1 indica que la relación es producto del azar
- **Lift > 1** indica una relación realmente fuerte (controlado por la frecuencia con que ambos ocurren)
- **Lift > 1** indica que A y B aparecen juntos con mas frecuencia de lo que indica el azar (complementos)
- **Lift < 1** indica una relación realmente débil (controlado por la frecuencia con que ambos ocurren)
- **Lift < 1** indica que A y B aparecen juntos con menos frecuencia de lo que indica el azar

Ejemplo Lift

- **Soporte**(Pan)=**0.70**
- **Soporte**(JN)=**0.40**
- **Soporte**(JN=>Pan) =
Soporte(Pan=>JN)=**0.28**
- **Confianza**(Pan=>JN)= $0.28/0.70$ =**0.40**
- **Confianza**(JN=>Pan)= $0.28/0.40$ =**0.70**

- **Lift**(Pan=>JN)=**Lift**(JN=>Pan)

Soporte(Pan=>JN)/Soporte(Pan)*Soporte(JN)

$$0.28/(0.70*0.40)=0.28/0.28=1$$

Ejemplo Lift

- **Soporte(JN)=0.40**
- **Soporte(V)=0.05**
- **Soporte(JN=>V) = Soporte(V=>JN)=0.04**
- **Confianza(V=>JN)=0.04/0.05=0.80**
- **Confianza(JN=>V)=0.04/0.40=0.10**

Lift(Vodka=>JN)=Lift(JN=>V)=?

Soporte(Vodka=>JN)/Soporte(JN)*Soporte(V)

$$0.04/(0.4*0.05)=0.04/0.02=2$$

Aprendizaje de Reglas de Asociación

- Deben establecerse los requisitos mínimos
Ej: soporte > 0.02
- Aprendizaje
 - Extracción del conjunto de items que cumple con el soporte requerido.
 - Generación de las reglas a partir de estos items.

Algoritmo *A priori*

- Identificar los items que en forma individual cumplen con el soporte mínimo.
- Utilizar estos items para formar conjuntos de dos items que cumplen con la cobertura mínima.
- Utilizar los items anteriores para formar grupos de a tres.
- Seguir hasta que no encontrar un grupo mayor que cumpla con los requisitos.

Algoritmo A priori - Importancia de los conjuntos de items frecuentes

- Hallar los *itemsets frecuentes*: conjuntos de items que tienen mínimo soporte
 - Un subconjunto de un itemset frecuente debe ser también un itemset frecuente
 - si $\{AB\}$ es *un* itemset frecuente, luego $\{A\}$ y $\{B\}$ deberían ser itemsets frecuentes
 - Iterativamente hallar los itemsets frecuentes con cardinalidad desde 1 a k (k -itemset)
- Usar los itemsets frecuentes para generar reglas de asociación.

Algoritmo Apriori (D:datos, MinC : cobertura mínima)

i = 0

Rellena_Item(C_i)

mientras $C_i \neq \emptyset$

para cada x = elemento de C_i

 Si Cobertura(x) \geq MinC **entonces** $L_i = L_i \cup x$

fin para

$C_{i+1} = \text{Selecciona_Candidatos}(L_i)$

i = i + 1

fin mientras

retorna C

Algoritmo A priori

(*sop.mín.=0.5*)

Productos: 1-Papas; 2-Leche; 3-Huevos; 4-Jamón; 5-Pan

Database D

TID	Items
Juan	1 3 4
María	2 3 5
Luis	1 2 3 5
Ana	2 5

Scan D

C_1

itemset	sup.
{1}	0.5
{2}	0.75
{3}	0.75
{4}	0.25
{5}	0.75

L_1

itemset	sup.
{1}	0.5
{2}	0.75
{3}	0.75
{5}	0.75

C_2

itemset	sup
{1 2}	0.25
{1 3}	0.5
{1 5}	0.25
{2 3}	0.5
{2 5}	0.75
{3 5}	0.5

Scan D

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

L_2

itemset	sup
{1 3}	0.5
{2 3}	0.5
{2 5}	0.75
{3 5}	0.5

C_3

itemset
{2 3 5}

Scan D

L_3

itemset	sup
{2 3 5}	0.5

Algoritmo *A priori*

(*sop.mín.* = 0.5)

24

Productos: 1-Papas; 2-Leche; 3-Huevos; 4-Jamón; 5-Pan

TID	Items
Juan	1 3 4
María	2 3 5
Luis	1 2 3 5
Ana	2 5

→ ... →

L_3

itemset	sup
{2 3 5}	0.5

- Una vez obtenido el conjunto de items frecuentes se forman las reglas y se analizan:
- Por ejemplo
 - Pan → Leche y huevos
 - Leche y huevos → Pan

Desventajas del algoritmo A priori

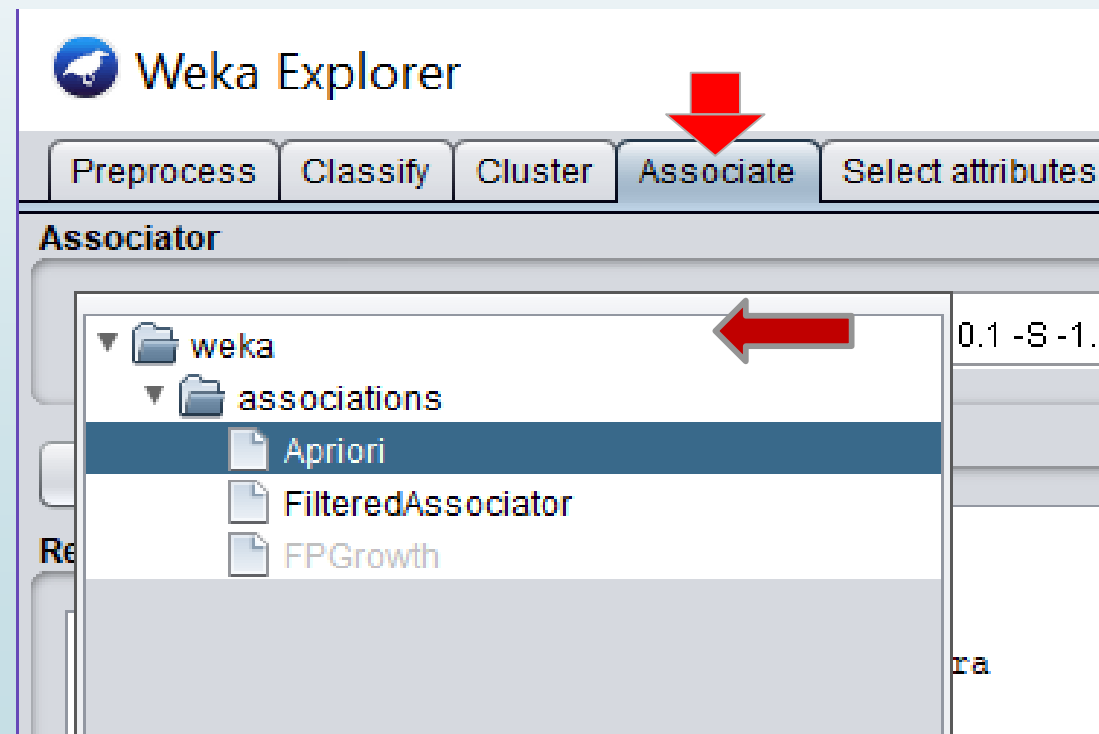
- Recorre la base de datos varias veces
 - Este es el proceso más costoso.
- Genera conjuntos de items frecuentes de cardinalidad alta.
 - Dificulta la construcción de las reglas llevando a verificar sobre la base de datos muchas opciones.

- Para mejorar la búsqueda de reglas de asociación se han propuesto variantes al algoritmo básico
 - Tablas hash
 - Uso de una estructura tipo árbol
Ej: Frequent Pattern Tree [Huan et al.2000]
 - Técnicas paralelización

A priori en Weka

Cargue en el panel Preprocess el archivo Golf_Nominal.csv, para determinar si se puede jugar dada unas condiciones climáticas determinadas.

Luego diríjase a la pestaña, Associate.



A priori en Weka

weka.gui.GenericObjectEditor

weka.associations.Apriori

Capabilities

car	False
classIndex	-1
delta	0.05
doNotCheckCapabilities	False
lowerBoundMinSupport	0.1
metricType	Lift
minMetric	1.1
numRules	10
outputItemSets	False
removeAllMissingCols	False
significanceLevel	-1.0
treatZeroAsMissing	False
upperBoundMinSupport	1.0
verbose	False

Apriori

=====

Best rules found:

1. Temperatura=baja 4 ==> Humedad=Normal 4 conf:(1) < lift:(2)> lev:(0.14) [2] conv:(2)
2. Humedad=Normal 7 ==> Temperatura=baja 4 conf:(0.57) < lift:(2)> lev:(0.14) [2] conv:(1.25)
3. Humedad=alta 7 ==> Juega=No 4 conf:(0.57) < lift:(1.6)> lev:(0.11) [1] conv:(1.13)
4. Juega=No 5 ==> Humedad=alta 4 conf:(0.8) < lift:(1.6)> lev:(0.11) [1] conv:(1.25)
5. Ambiente=nublado 4 ==> Juega=Si 4 conf:(1) < lift:(1.56)> lev:(0.1) [1] conv:(1.43)
6. Juega=Si 9 ==> Ambiente=nublado 4 conf:(0.44) < lift:(1.56)> lev:(0.1) [1] conv:(1.07)
7. Humedad=Normal Viento=NO 4 ==> Juega=Si 4 conf:(1) < lift:(1.56)> lev:(0.1) [1] conv:(1.43)
8. Juega=Si 9 ==> Humedad=Normal Viento=NO 4 conf:(0.44) < lift:(1.56)> lev:(0.1) [1] conv:(1.07)
9. Humedad=Normal 7 ==> Juega=Si 6 conf:(0.86) < lift:(1.33)> lev:(0.11) [1] conv:(1.25)
10. Juega=Si 9 ==> Humedad=Normal 6 conf:(0.67) < lift:(1.33)> lev:(0.11) [1] conv:(1.13)

Algoritmo FP-Growth

- Utiliza una estructura adicional con forma de árbol, denominada **FP-Tree (frequent pattern tree)**, que simplifica el acceso a la información.
- Recorre la estructura en forma recursiva determinando los conjuntos de ítems frecuentes con los que formará las reglas.

FP-Tree (Frequent Pattern Tree)

31

Construcción – Paso 1

- Ordenar los ítems según su frecuencia. Luego utilizar este orden para reescribir los ejemplos.

TID	Items
Juan	1 3 4
María	2 3 5
Luis	1 2 3 5
Ana	2 5

Item	#
1	2
2	3
3	3
4	1
5	3

FP-Tree (Frequent Pattern Tree)

32

Construcción – Paso 1

- Ordenar los ítems según su frecuencia. Luego utilizar este orden para reescribir los ejemplos.

TID	Items
Juan	1 3 4
María	2 3 5
Luis	1 2 3 5
Ana	2 5

Item	#
2	3
3	3
5	3
1	2
4	1

FP-Tree (Frequent Pattern Tree)

Construcción – Paso 1

- Ordenar los ítems según su frecuencia. Luego utilizar este orden para reescribir los ejemplos.

TID	Items
Juan	1 3 4
María	2 3 5
Luis	1 2 3 5
Ana	2 5

Item	#
2	3
3	3
5	3
1	2
4	1

TID	Items
Juan	3 1
María	2 3 5
Luis	2 3 5 1
Ana	2 5

FP-Tree (Frequent Pattern Tree)

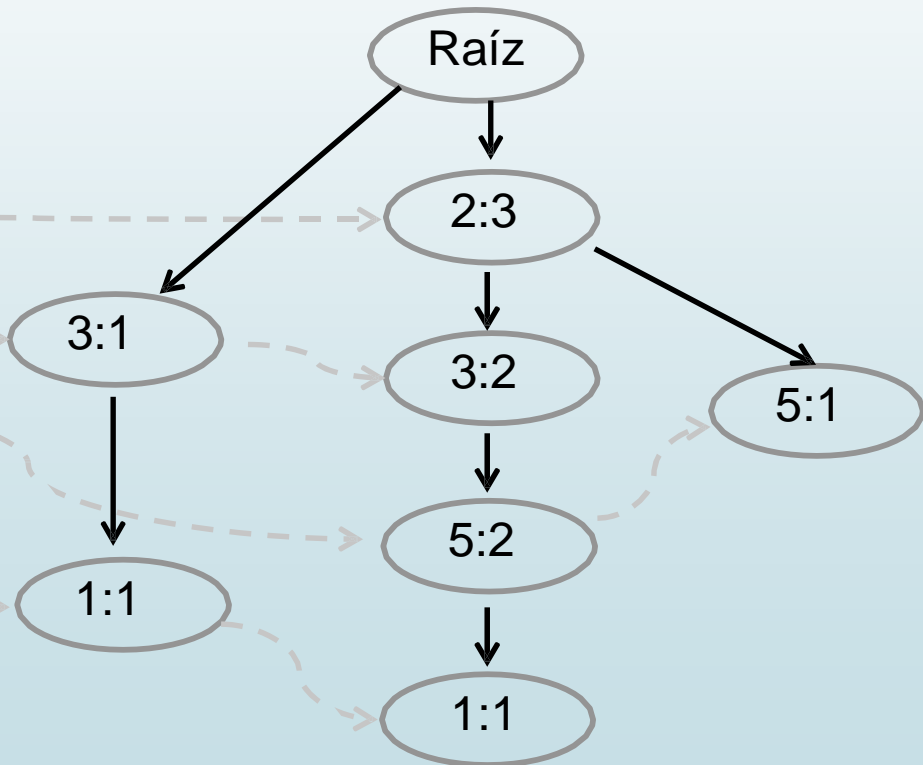
34

Construcción – Paso 2

- A partir de los ejemplos ordenados se construye el árbol

TID	Items
Juan	3 1
María	2 3 5
Luis	2 3 5 1
Ana	2 5

Item	inicio
2	
3	
5	
1	



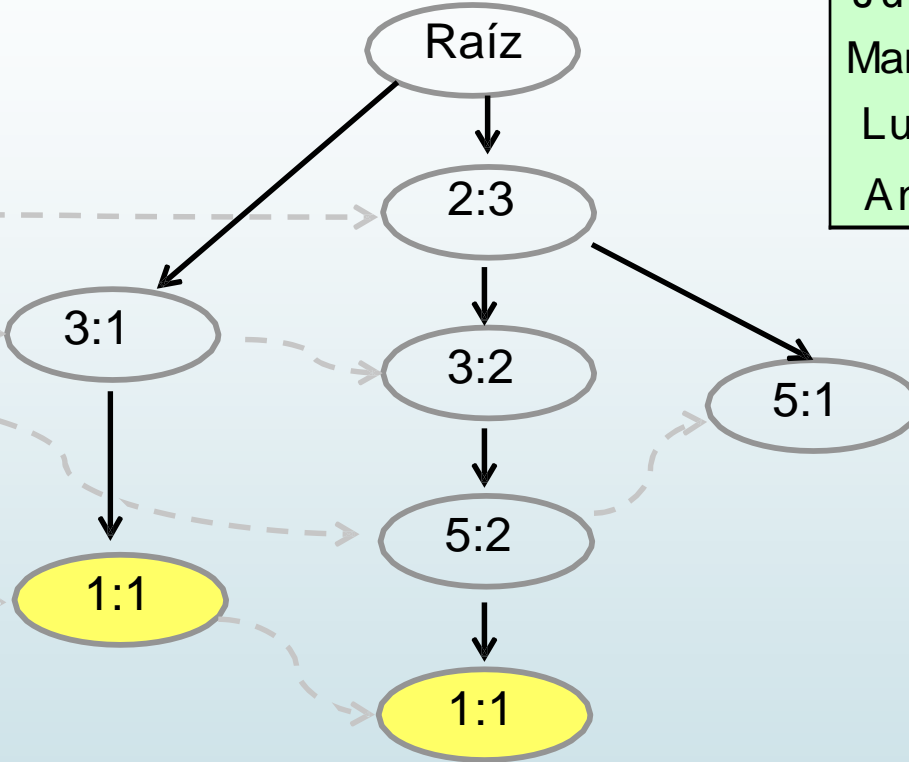
FP-Tree (Frequent Pattern Tree)

35

Construcción – Paso 3

TID	Items
Juan	3 1
María	2 3 5
Luis	2 3 5 1
Ana	2 5

Item	inicio
2	
3	
5	
1	



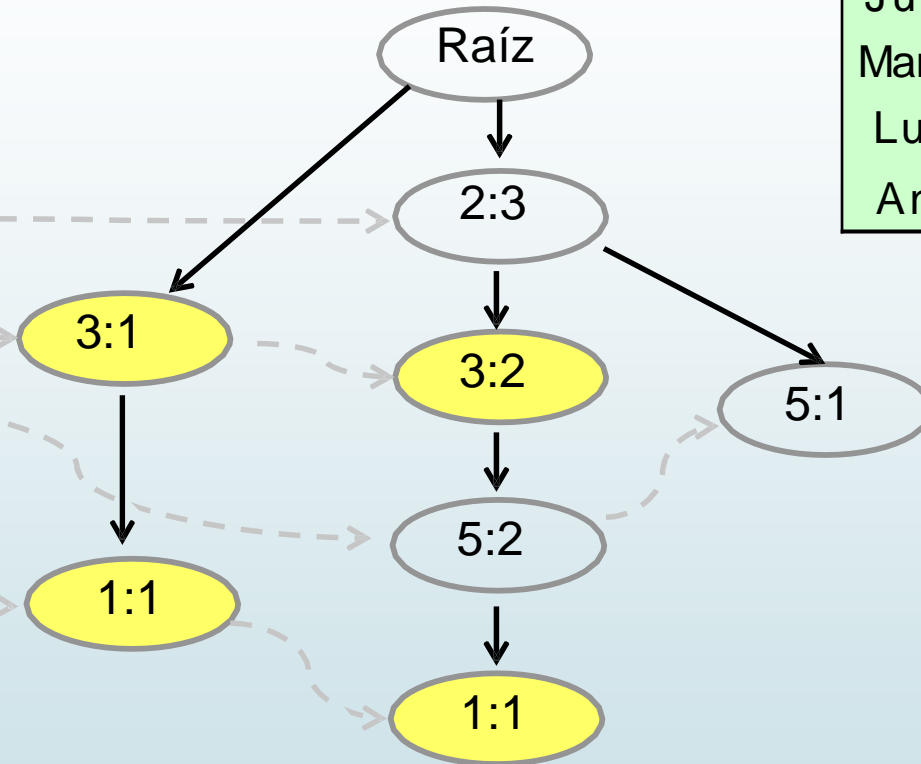
FP-Tree (Frequent Pattern Tree)

36

Construcción – Paso 3

TID	Items
Juan	3 1
María	2 3 5
Luis	2 3 5 1
Ana	2 5

Item	inicio
2	
3	
5	
1	



{1,3} → frecuencia 2

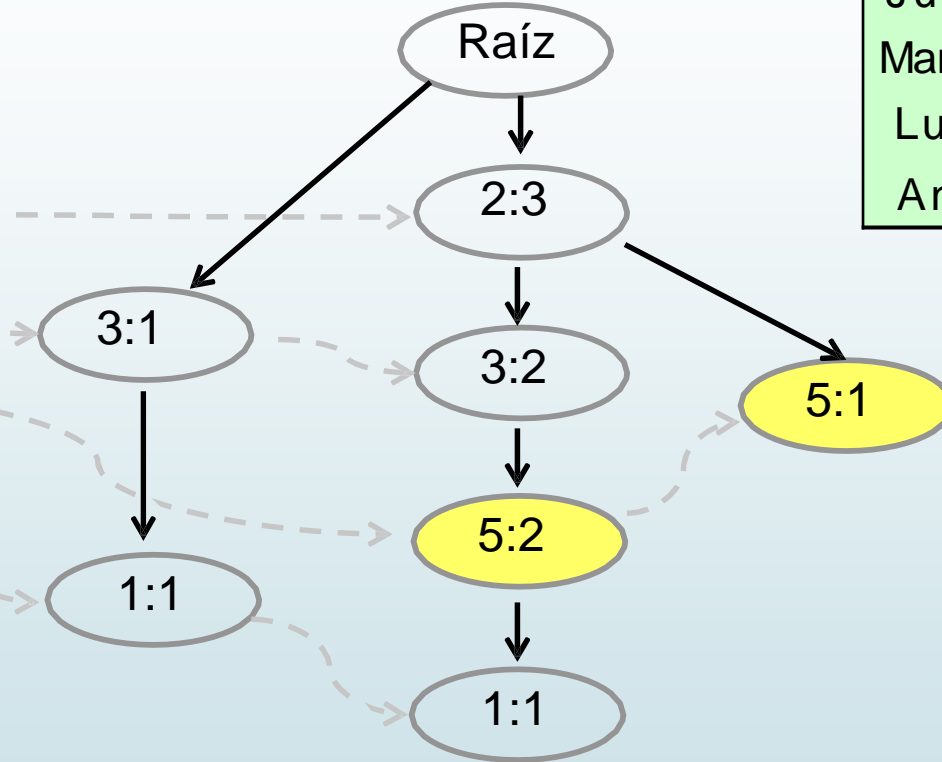
FP-Tree (Frequent Pattern Tree)

37

Construcción – Paso 3

TID	Items
Juan	3 1
María	2 3 5
Luis	2 3 5 1
Ana	2 5

Item	inicio
2	
3	
5	
1	



{1,3} → frecuencia 2

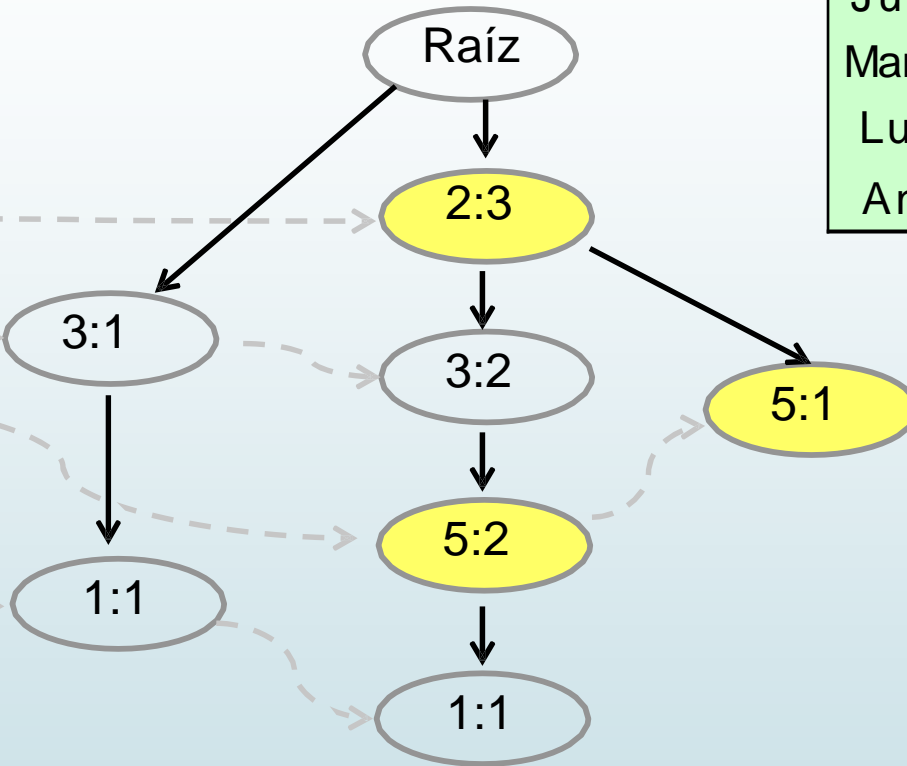
FP-Tree (Frequent Pattern Tree)

38

Construcción – Paso 3

TID	Items
Juan	3 1
María	2 3 5
Luis	2 3 5 1
Ana	2 5

Item	inicio
2	
3	
5	
1	



{1,3} → frecuencia 2

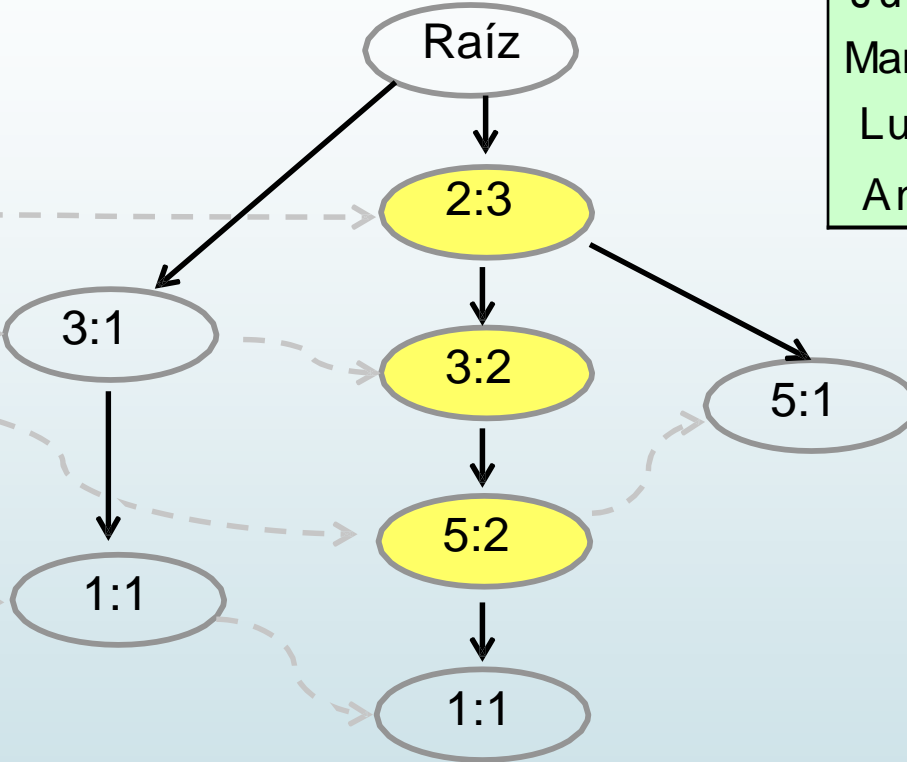
{2,5} → frecuencia 3

FP-Tree (Frequent Pattern Tree)

Construcción – Paso 3

TID	Items
Juan	3 1
María	2 3 5
Luis	2 3 5 1
Ana	2 5

Item	inicio
2	
3	
5	
1	



$\{1, 3\} \rightarrow$ frecuencia 2
 $\{2, 5\} \rightarrow$ frecuencia 3
 $\{2, 3, 5\} \rightarrow$ frecuencia 2

¿Otros pares con
frecuencia 2?

- Abrir el archivo **Compras.xlsx**

ID	Papas	Leche	Huevos	Jamon	Pan
Juan	1	0	1	1	0
Maria	0	1	1	0	1
Luis	1	1	1	0	1
Ana	0	1	0	0	1

- Abrir con Weka el archivo **Compras.xlsx**


Relation: WekaExcel

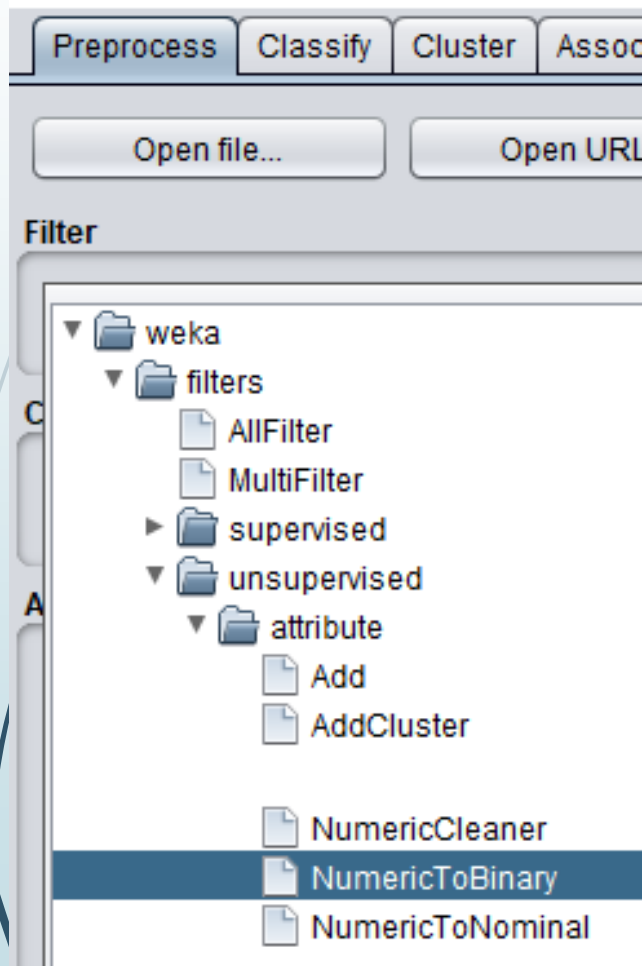
No.	1: ID	2: Papas	3: Leche	4: Huevos	5: Jamon	6: Pan
	Nominal	Numeric	Numeric	Numeric	Numeric	Numeric
1	Juan	1.0	0.0	1.0	1.0	0.0
2	Maria	0.0	1.0	1.0	0.0	1.0
3	Luis	1.0	1.0	1.0	0.0	1.0
4	Ana	0.0	1.0	0.0	0.0	1.0

■ Borrar el atributo ID

No.		Name
1	<input checked="" type="checkbox"/>	ID
2	<input type="checkbox"/>	Papas
3	<input type="checkbox"/>	Leche
4	<input type="checkbox"/>	Huevos
5	<input type="checkbox"/>	Jamon
6	<input type="checkbox"/>	Pan

Remove





- Luego convertir a binario

NumericToBinary

FP-Growth con Weka

The image shows the Weka GUI with the 'Preprocess' tab selected. A file tree on the left shows the path: weka > filters > unsupervised > attribute > NumericToBinary. The 'NumericToBinary' filter is selected. A dialog box titled 'weka.gui.GenericObjectEditor' is open, showing the 'About' section for 'weka.filters.unsupervised.attribute.NumericToBinary'. The dialog box contains the following text: 'Converts all numeric attributes into binary attributes (apart from the class attribute, if set): if the value of the numeric attribute is exactly zero, the value of the new attribute will be zero.' Below this, there are three settings: 'doNotCheckCapabilities' set to 'false', 'ignoreClass' set to 'True', and 'invertSelection' set to 'False'. A red arrow points to the 'doNotCheckCapabilities' dropdown. At the bottom of the dialog box, there are buttons for 'Open...', 'Save...', 'OK', and 'Cancel'. A red text box is overlaid on the dialog box with the text: 'Aplicar también al atributo de la clase'.

weka.gui.GenericObjectEditor

weka.filters.unsupervised.attribute.NumericToBinary

About

Converts all numeric attributes into binary attributes (apart from the class attribute, if set): if the value of the numeric attribute is exactly zero, the value of the new attribute will be zero.

More

Capabilities

Aplicar también al atributo de la clase

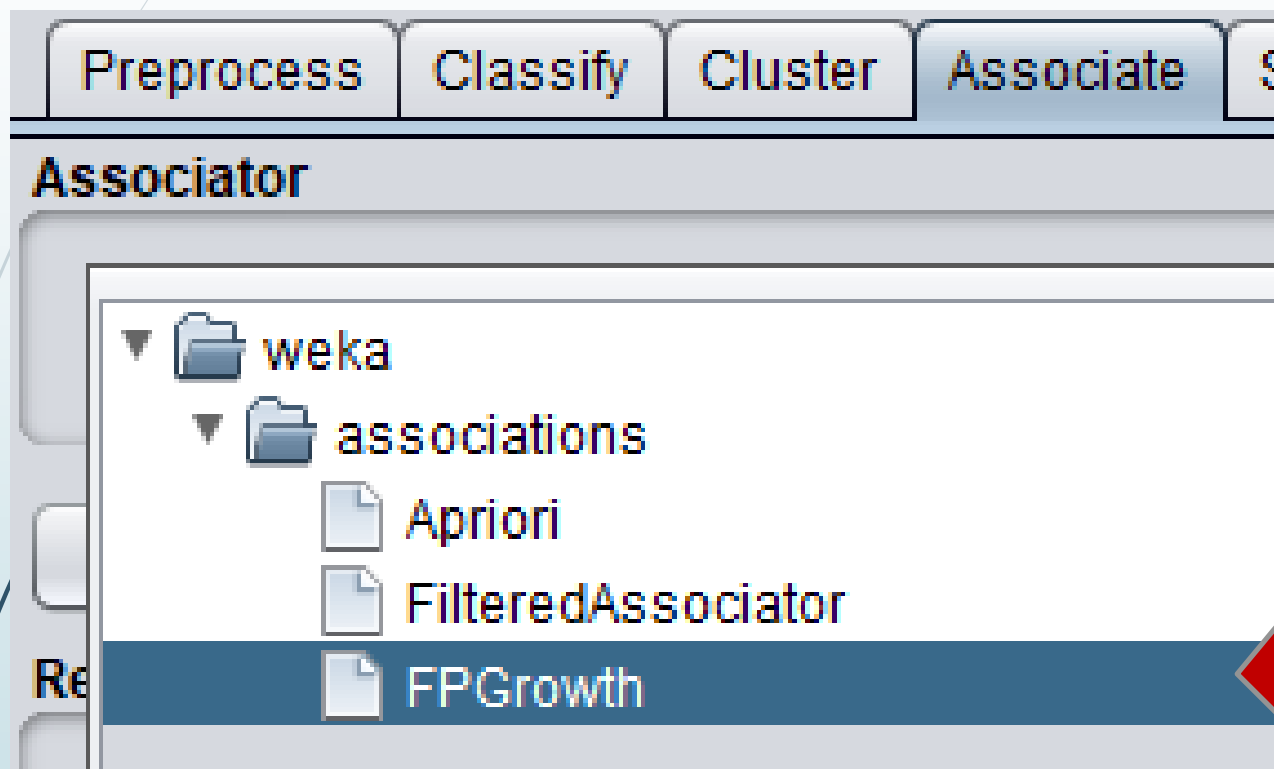
doNotCheckCapabilities false

ignoreClass True

invertSelection False

Open... Save... OK Cancel

FP-Growth con Weka



FPGrowth found 19 rules (displaying top 10)

1. [Pan_binarized=1]: 3 ==> [Leche_binarized=1]: 3 <conf:(1)> lift:(1.33) lev:(0.
2. [Leche_binarized=1]: 3 ==> [Pan_binarized=1]: 3 <conf:(1)> lift:(1.33) lev:(0.
3. [Papas_binarized=1]: 2 ==> [Huevos_binarized=1]: 2 <conf:(1)> lift:(1.33) lev:
4. [Jamon_binarized=1]: 1 ==> [Huevos_binarized=1]: 1 <conf:(1)> lift:(1.33) lev:
5. [Jamon_binarized=1]: 1 ==> [Papas_binarized=1]: 1 <conf:(1)> lift:(2) lev:(0.1
6. [Pan_binarized=1, Huevos_binarized=1]: 2 ==> [Leche_binarized=1]: 2 <conf:(1)>
7. [Leche_binarized=1, Huevos_binarized=1]: 2 ==> [Pan_binarized=1]: 2 <conf:(1)>
8. [Pan_binarized=1, Papas_binarized=1]: 1 ==> [Leche_binarized=1]: 1 <conf:(1)>
9. [Leche_binarized=1, Papas_binarized=1]: 1 ==> [Pan_binarized=1]: 1 <conf:(1)>
10. [Pan_binarized=1, Papas_binarized=1]: 1 ==> [Huevos_binarized=1]: 1 <conf:(1)>

Ejercicio 3 – Titanic.arff

Vamos a estudiar ahora los datos del hundimiento del Titanic. Los datos se encuentran en el archivo "titanic.arff" y corresponden a las características de los 2.201 pasajeros del Titanic. Estos datos son reales y se han obtenido de: "Report on the Loss of the 'Titanic' (S.S.)" (1990), British Board of Trade Inquiry Report_ (reprint), Gloucester, UK: Allan Sutton Publishing. Para este ejemplo sólo se van a considerar cuatro variables:

- Clase (0 = tripulación, 1 = primera, 2 = segunda, 3 = tercera)
- Edad (1 = adulto, 0 = niño)
- Sexo (1 = hombre, 0 = mujer)
- Sobrevivió (1 = sí, 0 = no)

Analizar e interpretar que reglas de asociación podemos extraer de estos atributos.