

Reglas de Asociación con Knime



**UNIVERSIDAD
DE GRANADA**

Pablo Morenilla Pinos
morenillapablo@correo.ugr.es
TID Prácticas Grupo 1

Índice

1. Market (6 puntos)	3
2. Cesta de la compra (4 puntos)	14

Índice de figuras

1.	Lectura del fichero CSV market.	3
2.	Cambio del tipo entero a Sting en la lectura del fichero market.	3
3.	Configuración del nodo Apriori.	4
4.	Resultados obtenidos.	4
5.	Cambiar 0 a null.	6
6.	Integer to String.	6
7.	Cambiar 1 a su nombre.	7
8.	Función del nodo.	7
9.	Nodo Create Collection Column.	8
10.	Nodo Create Collection demostración resultado.	8
11.	Configuración nodo item set finder.	9
12.	Resultados de mayor frecuencia con un tamaño de 2.	9
13.	Resultados de mayor frecuencia con un tamaño de 3.	10
14.	Configuración del nodo Association Rule Learner.	10
15.	Mejores resultados de tamaño 2 con una confianza.	10
16.	Algunos resultados con confianza rondando 0.6.	10
17.	Configuración del nodo Rule Viewer.	11
18.	Demostración gráfica del rule viewer con los datos.	11
19.	Lectura del fichero datos compras con File Reader.	14
20.	Delimitar los datos por ','.	15
21.	Creación columna elementos comunes con Create Collection Column.	15
22.	Borgelt de los productos de la compra	16
23.	Implicaciones entre elementos de la compra.	16
24.	Visualización de los datos filtrados.	17
25.	If/else para crear columnas.	20
26.	Nodo Apriori con algunos datos de la compra.	20
27.	Nodo Association Rule Learner con algunos datos de la compra.	22
28.	Nodo Association Rule Learner ejemplos.	22
29.	Representación de la dispersión de los datos.	23
30.	Configuración del nodo Missing Value para No_Item_X.	24
31.	Resultado del uso del nodo Missing Value para No_Item_X.	24
32.	Implicaciones entre productos en el caso No_Item_X.	24
33.	Representación de la dispersión en el caso No_Item_X.	25

1. Market (6 puntos)

Considera el dataset en el fichero `market.csv`, que contiene la cesta de la compra de clientes. Realiza un estudio similar al explicado anteriormente, y encuentra reglas de asociación que establezcan co-ocurrencia entre productos.

a) Lea la base de datos en formato CSV.

Usamos el nodo **CSV Reader** y establecemos una serie de criterios para poder leer de manera adecuada el CSV, ya que por defecto agruparía todo en una sola columna, ya que no detecta bien el limitador para saltar de línea y crear columnas nuevas, para ello se le indica que salte de línea cuando esté el carácter `'`:

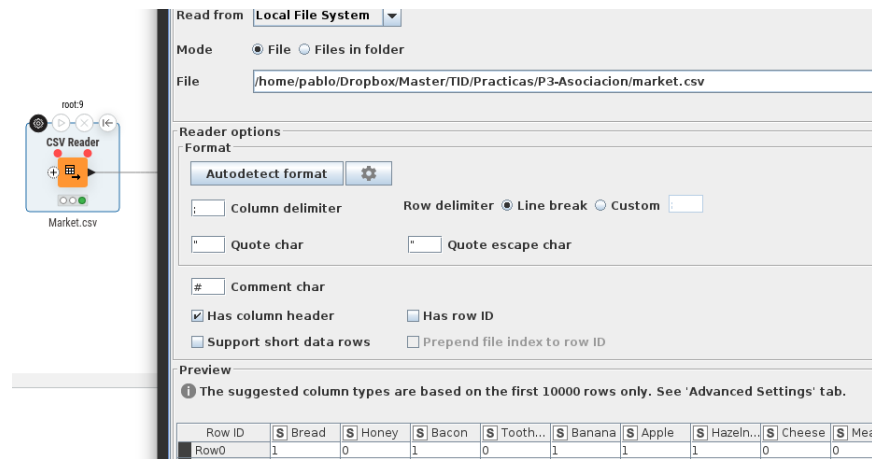


Figura 1: Lectura del fichero CSV market.

Después le indicamos en el apartado de transformación que los datos que está leyendo los interprete como tipo String para que, posteriormente, con el nodo apriori pueda leer bien los datos y para futuras operaciones que se quieren hacer con los datos (por defecto los interpreta como valores enteros):

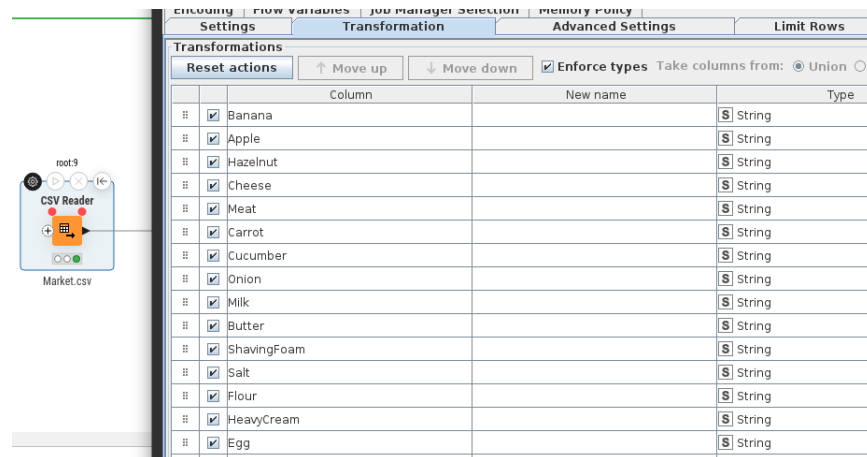


Figura 2: Cambio del tipo entero a Sting en la lectura del fichero market.

- b) La base de datos está en forma relacional, donde cada atributo tiene valor 1 o 0, según esté o no en la cesta de la compra. Sobre bases de datos relacionales, puede aplicar los nodos Apriori (3.7) o FPGrowth (3.7). Observará que las mejores reglas encontradas son entre productos que no se han comprado (valor 0).

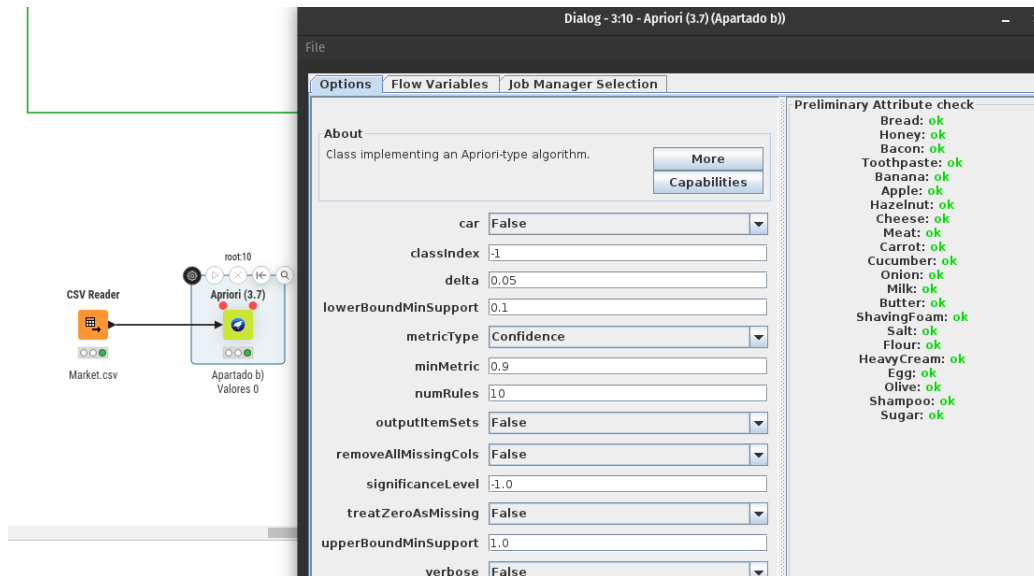


Figura 3: Configuración del nodo Apriori.

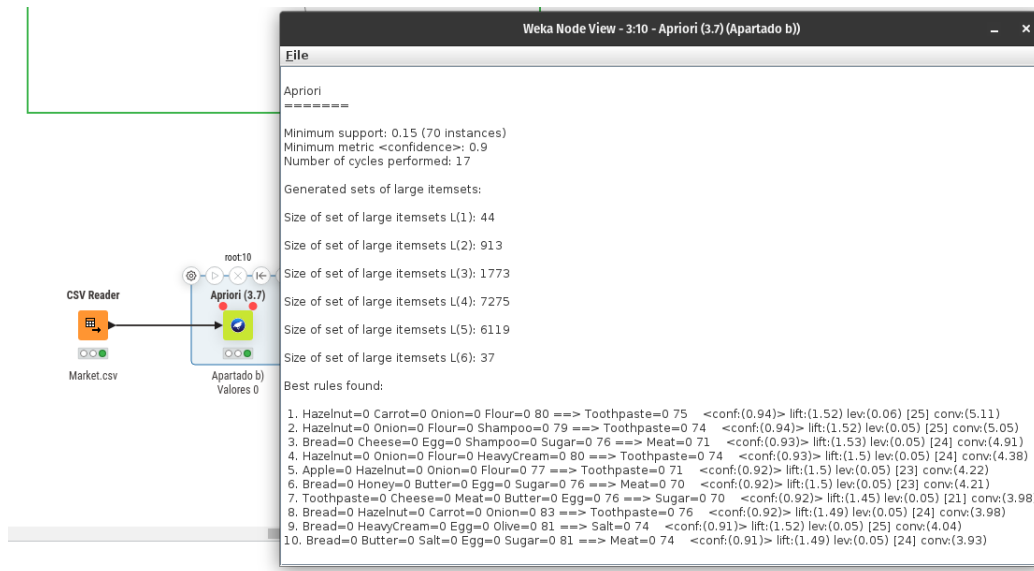


Figura 4: Resultados obtenidos.

El resultado final sería el de la imagen que muestra ese listado, destacando que se puede ver que los mejores resultados son cuando tienen los valores a 0.

■ **Best rules found:**

- a) Hazelnut=0 Carrot=0 Onion=0 Flour=0 80 \Rightarrow Toothpaste=0 75
;conf:(0.94);lift:(1.52) lev:(0.06) [25] conv:(5.11)
 - b) Hazelnut=0 Onion=0 Flour=0 Shampoo=0 79 \Rightarrow Toothpaste=0 74
;conf:(0.94);lift:(1.52) lev:(0.05) [25] conv:(5.05)
 - c) Bread=0 Cheese=0 Egg=0 Shampoo=0 Sugar=0 76 \Rightarrow Meat=0 71
;conf:(0.93);lift:(1.53) lev:(0.05) [24] conv:(4.91)
 - d) Hazelnut=0 Onion=0 Flour=0 HeavyCream=0 80 \Rightarrow Toothpaste=0 74
;conf:(0.93);lift:(1.5) lev:(0.05) [24] conv:(4.38)
 - e) Apple=0 Hazelnut=0 Onion=0 Flour=0 77 \Rightarrow Toothpaste=0 71
;conf:(0.92);lift:(1.5) lev:(0.05) [23] conv:(4.22)
 - f) Bread=0 Honey=0 Butter=0 Egg=0 Sugar=0 76 \Rightarrow Meat=0 70
;conf:(0.92);lift:(1.5) lev:(0.05) [23] conv:(4.21)
 - g) Toothpaste=0 Cheese=0 Meat=0 Butter=0 Egg=0 76 \Rightarrow Sugar=0 70
;conf:(0.92);lift:(1.45) lev:(0.05) [21] conv:(3.98)
 - h) Bread=0 Hazelnut=0 Carrot=0 Onion=0 83 \Rightarrow Toothpaste=0 76
;conf:(0.92);lift:(1.49) lev:(0.05) [24] conv:(3.98)
 - i) Bread=0 HeavyCream=0 Egg=0 Olive=0 81 \Rightarrow Salt=0 74
;conf:(0.91);lift:(1.52) lev:(0.05) [25] conv:(4.04)
 - j) Bread=0 Butter=0 Salt=0 Egg=0 Sugar=0 81 \Rightarrow Meat=0 74
;conf:(0.91);lift:(1.49) lev:(0.05) [24] conv:(3.93)
- c) Para encontrar reglas entre productos que sí se han comprado (valor 1), debe transformar la base de datos relacional en una transaccional. Es decir, debe definir una columna que describa la transacción. Por ejemplo, para una instancia que contenga todo ceros, excepto **Bread** = 1 y **Bacon** = 1, la transacción sería {**Bread**, **Bacon**}. Una vez tenga construida la base de datos transaccional, puede aplicar los nodos **Item Set Finder (Borgelt)** y **Association Rule Learner** para encontrar itemsets frecuentes y reglas de asociación importantes, respectivamente. Puede construir las transacciones como desee, una opción sería convertir las variables a tipo string y aplicar repetidamente el nodo **String Manipulation** para sustituir "1" por el nombre de cada atributo. A continuación, se puede convertir el valor "0" a valor perdido con el nodo **String Manipulation (Multi Column)** y crear las transacciones ignorando los valores perdidos con el nodo **Create Collection Column**.

Para este apartado usamos de nuevo el nodo **CSV Reader**, ya que ahora no queremos transformar directamente los datos tipo String, sino que, vamos a hacer una condición con el nodo 'Math Formular Multi Column' para asignarle a todas las columnas que tengan algún valor de 0 a nulo y para ello tienen que ser de tipo entero, luego:

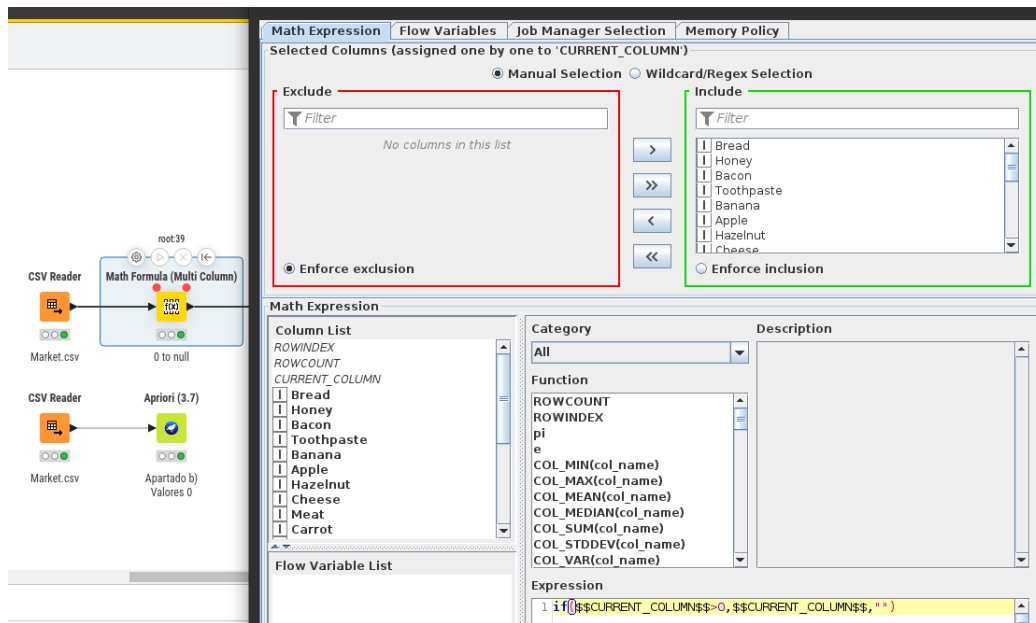


Figura 5: Cambiar 0 a null.

Como se puede ver, establecemos la condición de que si el valor que contiene la columna es mayor que cero, se deja el valor que tuviera, pero en caso contrario, se le asigna nulo.

Después, cambiamos el tipo entero a tipo string:

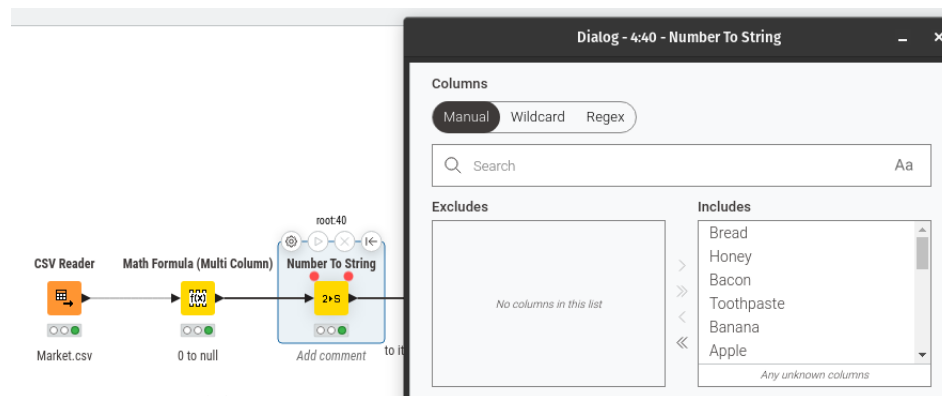


Figura 6: Integer to String.

Ahora creamos un metanodo que contiene todos los nodos encargados de reasignar a cada columna el nombre de la variable en lugar de tener asignado 1:

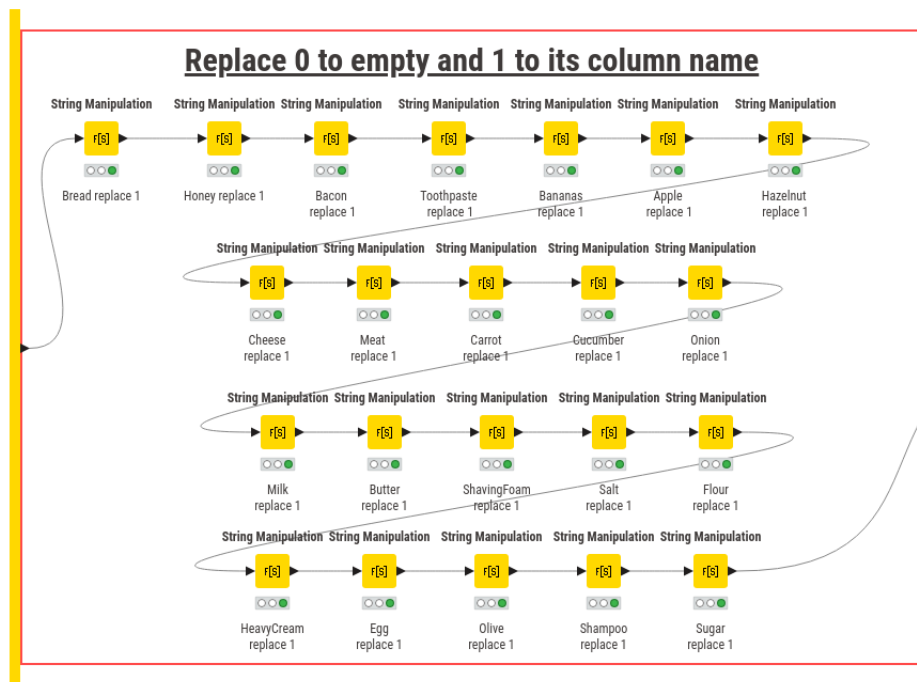


Figura 7: Cambiar 1 a su nombre.

Como se puede ver, establecemos con la función replace que, si el valor que hay en esa columna es "1", se cambia por el valor del nombre de su columna y se hacen esos cambios en la misma columna.

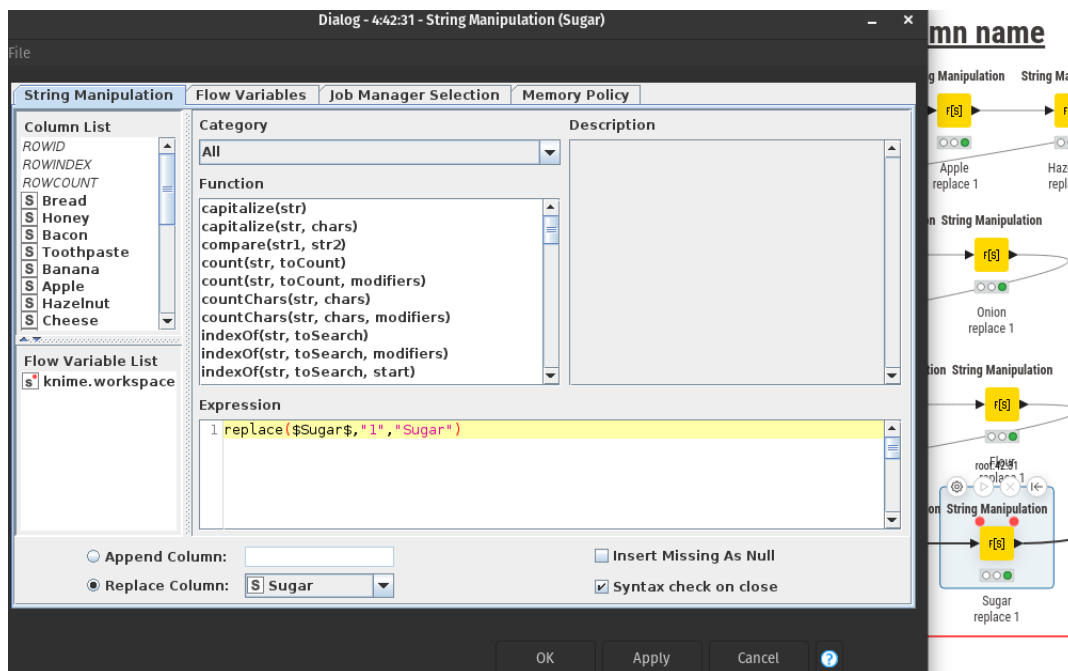


Figura 8: Función del nodo.

Por último, usamos el nodo Create Collection Column, para agrupar en una columna todos los valores que tengan en común por cada fila, así quedando finalmente solo los valores que sean a 1, pero con su nombre respectivamente, para que así sea por ejemplo {Bread, Bacon}

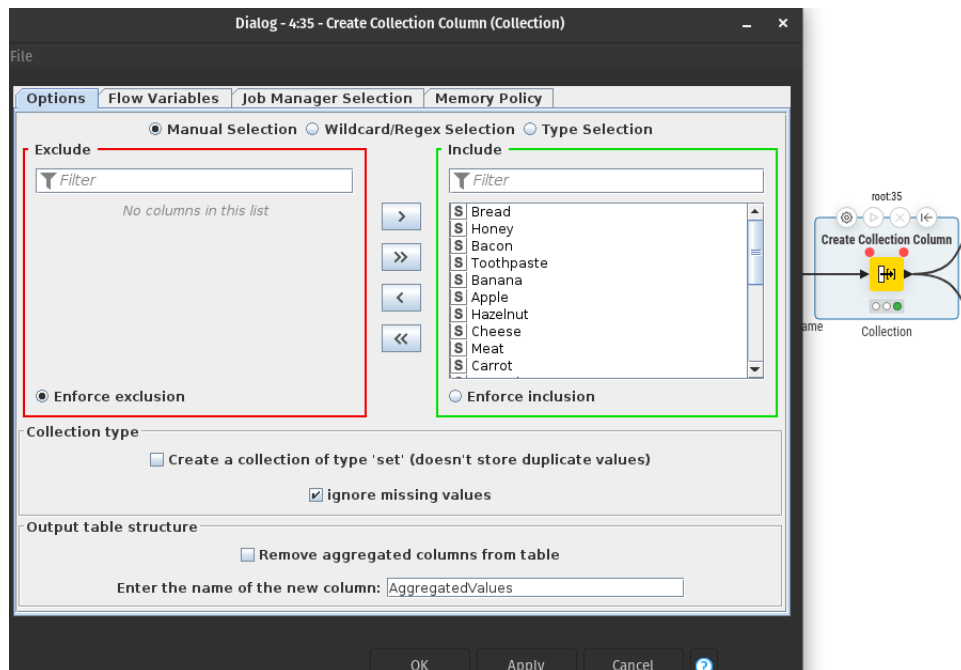


Figura 9: Nodo Create Collection Column.

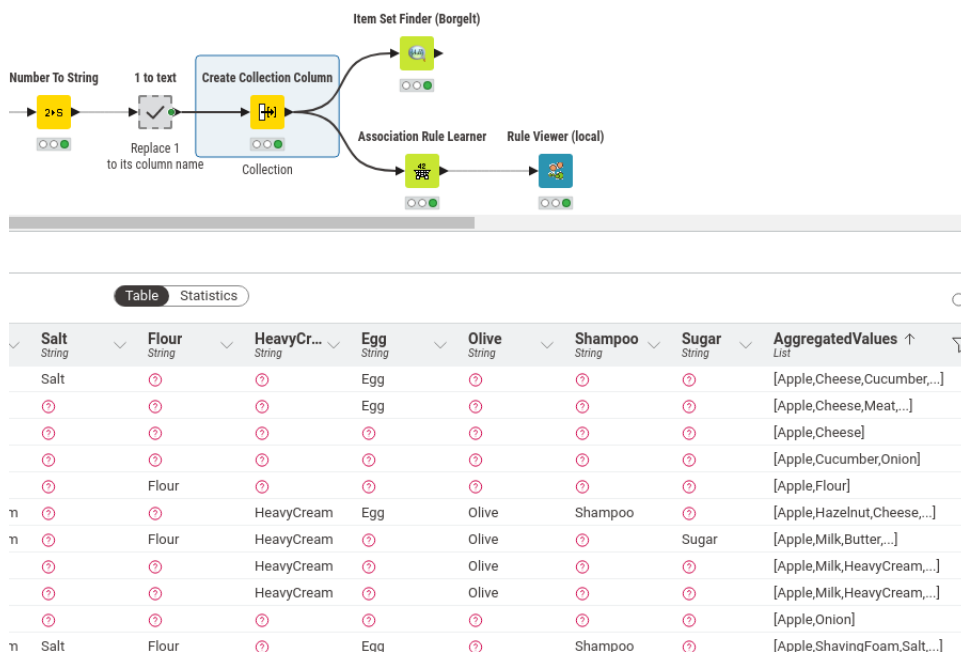


Figura 10: Nodo Create Collection demostración resultado.

- d) Identifique itemsets frecuentes y reglas de asociación importantes. Ajuste el soporte y confianza mínimos para obtener reglas que contengan al menos 2 ítems en el antecedente.

Usando el nodo Item set Finder (Borgelt) podemos encontrar los itemsets más frecuentes, para este caso y como dice el enunciado, se hará un filtrado para buscar los itemsets de al menos 2:

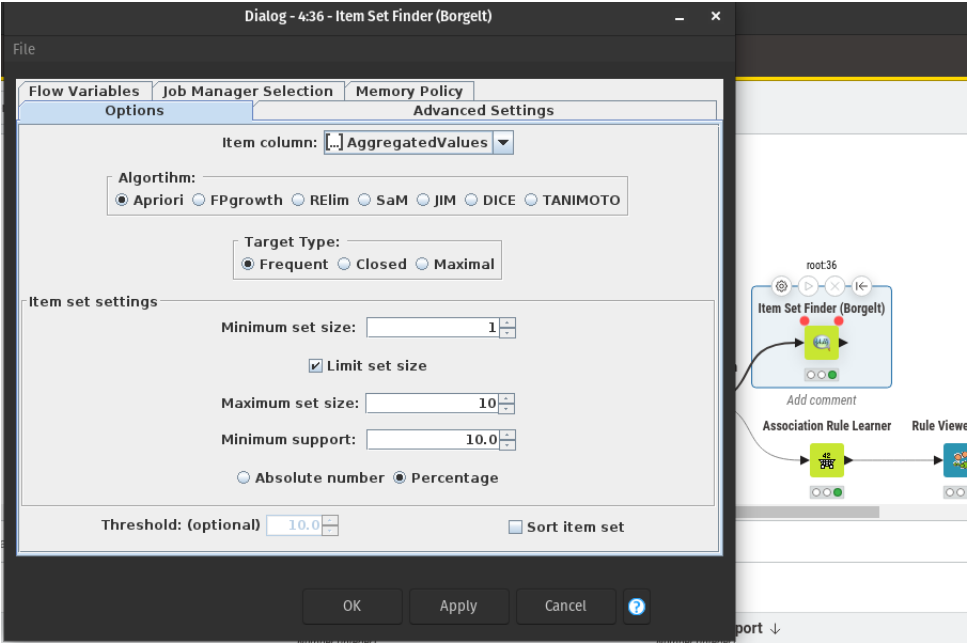


Figura 11: Configuración nodo item set finder.

#	RowID	ItemSet Set	ItemsetSize Number (integer)	ItemSetSupport Number (integer)	RelativeItemSetSupport% Number (double)
	RowID	ItemSet	2	ItemSetSupport	RelativeItemSetSupport%
546	Row545	[Bacon,Bananas]	2	112	24.138
544	Row543	[Bacon,Cheese]	2	104	22.414
548	Row547	[Cheese,Bananas]	2	104	22.414
586	Row485	[Egg,Cheese]	2	103	22.198
537	Row536	[Hazelnut,Bacon]	2	103	22.198
542	Row541	[Hazelnut,Bananas]	2	103	22.198
548	Row347	[Apple,Bananas]	2	102	21.983
564	Row463	[ShavingFoam,Bananas]	2	102	21.983
509	Row508	[Carrot,Bananas]	2	102	21.983
507	Row506	[Carrot,Cheese]	2	101	21.767
522	Row521	[Honey,Cheese]	2	101	21.767
531	Row230	[Butter,Cheese]	2	100	21.552

Figura 12: Resultados de mayor frecuencia con un tamaño de 2.

#	RowID	ItemSet Set	ItemsetSize Number (integer)	ItemSetSupport Number (integer) ↓	RelativeItemSetSupport% Number (double)
	RowID	ItemSet		ItemSetSupport	RelativeItemSetSupport%
545	Row544	[Bacon,Cheese,Bananas]	3	65	14.009
229	Row228	[Butter,Bacon,Cheese]	3	64	13.793
461	Row460	[ShavingFoam,Bacon,Bananas]	3	62	13.362
270	Row269	[Onion,Bacon,Bananas]	3	61	13.147
484	Row483	[Egg,Bacon,Cheese]	3	61	13.147
487	Row486	[Egg,Cheese,Bananas]	3	60	12.931
212	Row211	[Butter,ShavingFoam,Bacon]	3	59	12.716
434	Row433	[Bread,Bacon,Bananas]	3	59	12.716
460	Row459	[ShavingFoam,Bacon,Cheese]	3	59	12.716
508	Row507	[Carrot,Cheese,Bananas]	3	59	12.716
521	Row520	[Honey,Bacon,Bananas]	3	59	12.716
539	Row538	[Hazelnut,Bacon,Bananas]	3	59	12.716

Figura 13: Resultados de mayor frecuencia con un tamaño de 3.

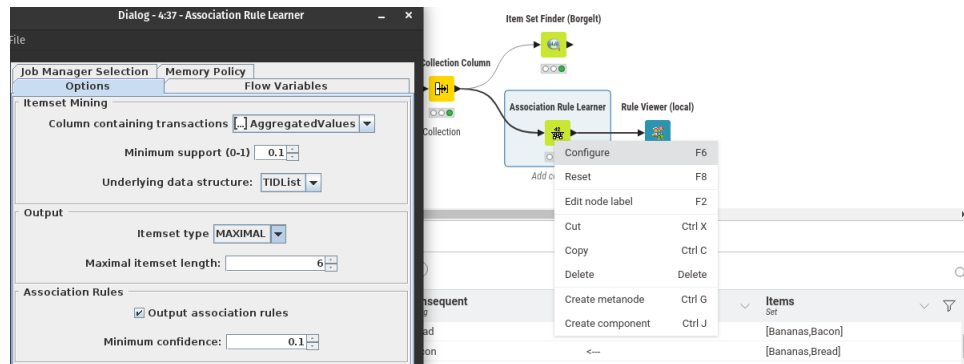


Figura 14: Configuración del nodo Association Rule Learner.

#	Row...	Support Number (double)	Confidence Number (double) ↓	Lift Number (double) ↓	Consequent String	Implies String	Items Set
191	rule1...	0.131	0.701	1.564	Bananas	<--	[Onion,Bacon]
612	rule6...	0.119	0.688	1.534	Bananas	<--	[Olive,ShavingFoam]
175	rule1...	0.121	0.675	1.565	Bacon	<--	[Bananas,Butter]
351	rule3...	0.127	0.67	1.555	Bacon	<--	[Butter,ShavingFoam]
344	rule3...	0.138	0.667	1.502	Cheese	<--	[Butter,Bacon]
922	rule9...	0.106	0.653	1.472	Cheese	<--	[Carrot,Butter]
1049	rule1...	0.106	0.653	1.472	Cheese	<--	[Egg,Butter]
29	rule28	0.121	0.651	1.511	Bacon	<--	[ShavingFoam,Bread]
1237	rule1...	0.116	0.651	1.465	Cheese	<--	[Onion,Butter]
295	rule2...	0.131	0.649	1.462	Cheese	<--	[Egg,Bacon]
1234	rule1...	0.106	0.645	1.452	Cheese	<--	[Butter,Meat]
149	rule1...	0.121	0.644	1.436	Bananas	<--	[Apple,Bacon]
538	rule5...	0.112	0.642	1.432	Bananas	<--	[Egg,Onion]
738	rule7...	0.108	0.641	1.525	Hazelnut	<--	[HeavyCream,Egg]

Figura 15: Mejores resultados de tamaño 2 con una confianza.

#	Row...	Support Number (double)	Confidence Number (double)	Lift Number (double)	Consequent String	Implies String	Items Set
	Row...	Support	Confidence	Lift	Consequent	Implies	Items
149	rule1...	0.121	0.644	1.436	Bananas	<--	[Apple,Bacon]
34	rule33	0.108	0.617	1.377	Bananas	<--	[Apple,Bread]
401	rule4...	0.116	0.614	1.369	Bananas	<--	[Apple,Carrot]
668	rule6...	0.114	0.602	1.494	Egg	<--	[Apple,Carrot]
428	rule4...	0.11	0.6	1.338	Bananas	<--	[Apple,Onion]
431	rule4...	0.11	0.6	1.338	Bananas	<--	[Apple,ShavingFoam]
3	rule2	0.127	0.628	1.4	Bananas	<--	[Bacon,Bread]
170	rule1...	0.116	0.635	1.417	Bananas	<--	[Bacon,Cucumber]
173	rule1...	0.114	0.609	1.359	Bananas	<--	[Bacon,Milk]
271	rule2...	0.116	0.621	1.492	HeavyCream	<--	[Bacon,Milk]
309	rule3...	0.119	0.632	1.63	Meat	<--	[Bacon,Sugar]
2	rule1	0.127	0.602	1.397	Bacon	<--	[Bananas,Bread]
175	rule1...	0.121	0.675	1.565	Bacon	<--	[Bananas,Butter]

Figura 16: Algunos resultados con confianza rondando 0.6.

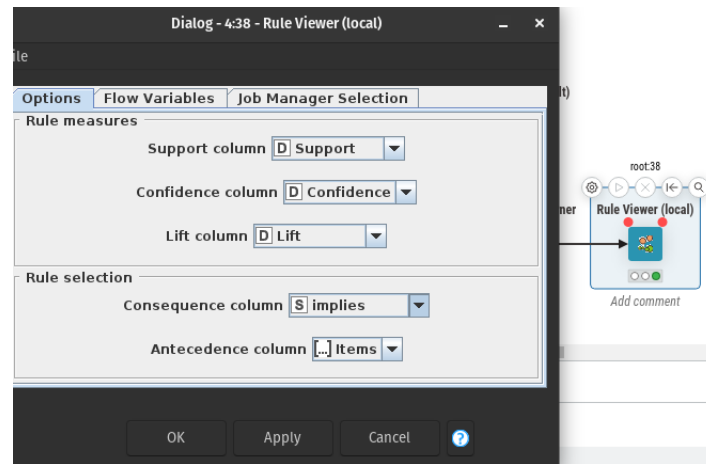


Figura 17: Configuración del nodo Rule Viewer.

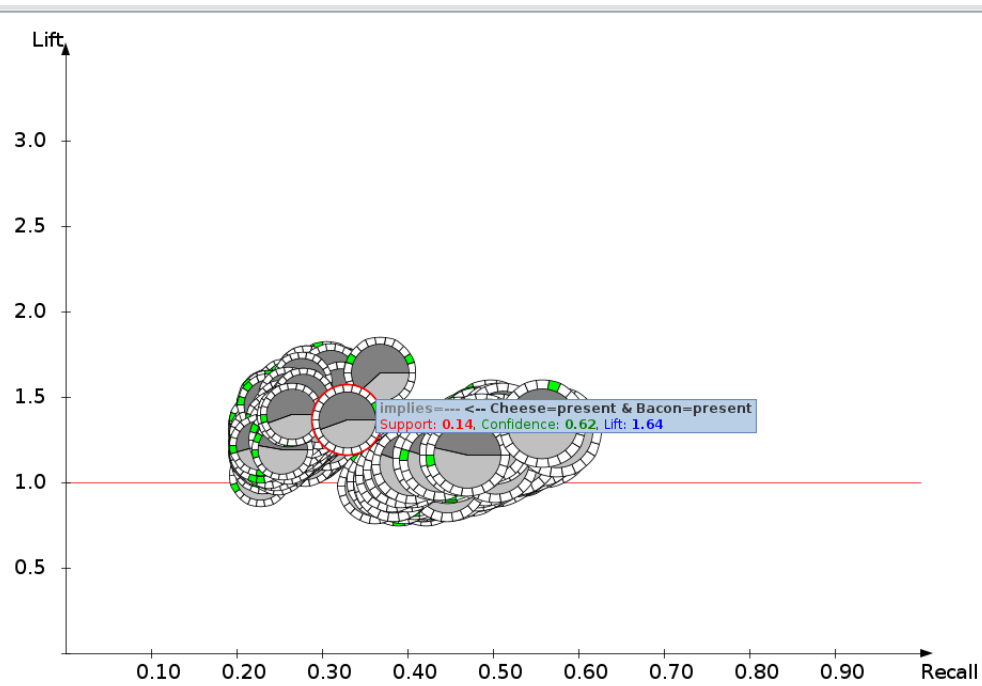


Figura 18: Demostración gráfica del rule viewer con los datos.

e) Discuta las relaciones interesantes que encuentre.

Tras lo visto en la imagen del apartado anterior, vamos a observar los 5 mejores resultados:

a) **Regla 1:**

- Confianza: 0.602
- Lift: 1.397
- Producto: Bacon
- Asociación: [Bananas, Bread]

b) **Regla 2:**

- Confianza: 0.527
- Lift: 1.293
- Producto: Bread
- Asociación: [Bananas, Bacon]

c) **Regla 3:**

- Confianza: 0.628
- Lift: 1.4
- Producto: Bananas
- Asociación: [Bacon, Bread]

d) **Regla 4:**

- Confianza: 0.563
- Lift: 1.307
- Producto: Bacon
- Asociación: [Hazelnut, Bread]

e) **Regla 5:**

- Confianza: 0.521
- Lift: 1.24
- Producto: Hazelnut
- Asociación: [Bacon, Bread]

De esto podemos obtener la siguiente información/análisis de los datos

- a)* **Productos Comunes en las Transacciones:** Productos como Bacon, Bread y Bananas son frecuentes en las transacciones, ya que aparecen en varias reglas de asociación.

-
- b) **Asociaciones Positivas Notables:** Se observan varias asociaciones positivas significativas, como la compra conjunta de Bacon con Bananas y Bread, indicando que estos productos tienden a ser adquiridos juntos en una proporción considerable de las transacciones.
 - c) **Combinaciones Específicas:** Algunas reglas destacan combinaciones específicas de productos, como Bacon, Bread y Bananas, sugiriendo preferencias específicas o hábitos de compra.
 - d) **Variedad en las Asociaciones:** Las reglas muestran una variedad de asociaciones, lo que sugiere que los clientes tienen opciones diversas en la composición de su cesta de la compra.
 - e) **Impacto de Productos Adicionales:** Algunas reglas indican que la presencia de ciertos productos adicionales, como Hazelnut, HeavyCream, Sugar, Butter y ShavingFoam, puede influir en las compras de productos más comunes.
 - f) **Fuerza de las Asociaciones:** Se observa variabilidad en la fuerza de las asociaciones, medida por la confianza y el lift. Algunas asociaciones, como Bread con Butter y Bacon, tienen confianza y lift altos, indicando fuertes vínculos de co-ocurrencia.

Por otro lado, si analizamos los resultados obtenidos en el nodo Item Ser Finder(Borgelt), en el caso de los item con un tamaño de 3, destacan [Bacon,Cheese,Bananas], [Butter,Bacon,Cheese], [Egg,Bacon,Cheese], [Butter,Bread,Cheese] entre otros, destacando así que cada vez que se realiza la compra, al menos cuando se da el caso de un producto como es el del Bacon, tiene bastante probabilidad de que se compre también otro como es con el Cheese, Egg, Butter, o Bread.

En el caso de los productos más frecuentes que tienen un itemsize de dos, se vuelve a ver el caso anterior pero también destaca que, hay casos como el de [Carrot,Bananas], [Apple,Bananas] donde la compra de una fruta se asocia a otra por lo que es frecuente la compra de varias frutas.

2. Cesta de la compra (4 puntos)

Considera el dataset en el fichero `datos_compras.csv`, que contiene la cesta de la compra de clientes de un supermercado. En este caso, el fichero contiene las transacciones en cada fila.

- a) Lea los datos y considere realizar cierto preprocesamiento. Para la lectura puede utilizar el nodo **File Reader (Complex Format)**, que generará una columna tipo string con los items delimitados por comas. A continuación, puede dividir en celdas diferentes con la coma como delimitador (nodo **Cell Splitter**), y crear una columna con los itemsets mediante el nodo **Create Collection Column**, ignorando los valores perdidos.

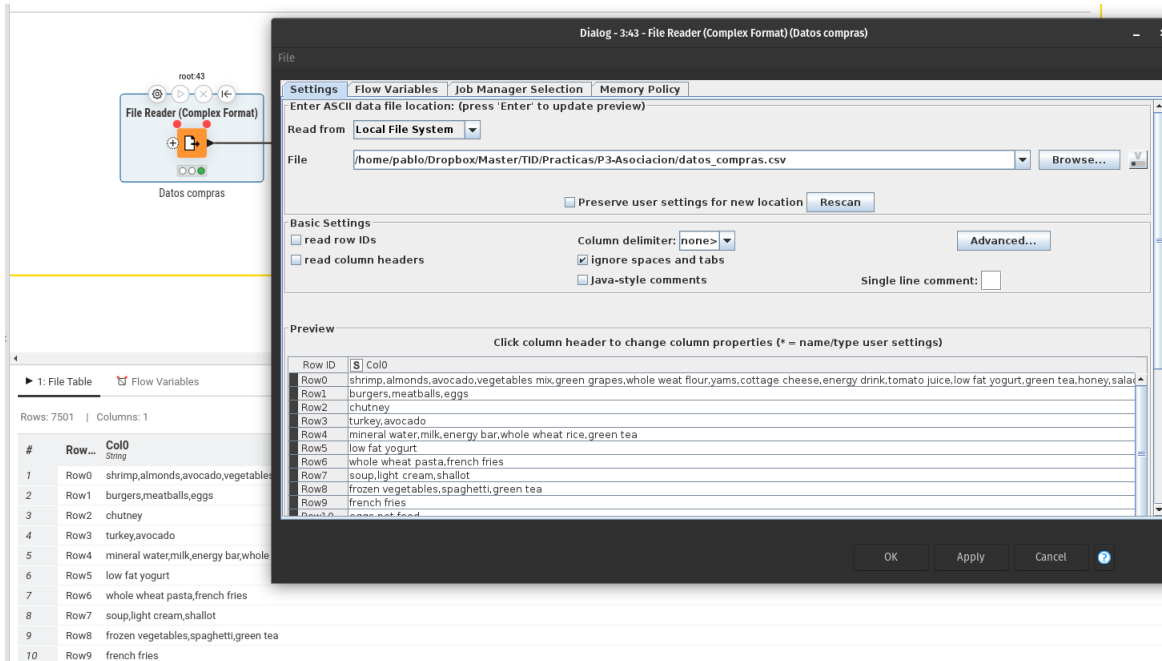


Figura 19: Lectura del fichero datos compras con File Reader.

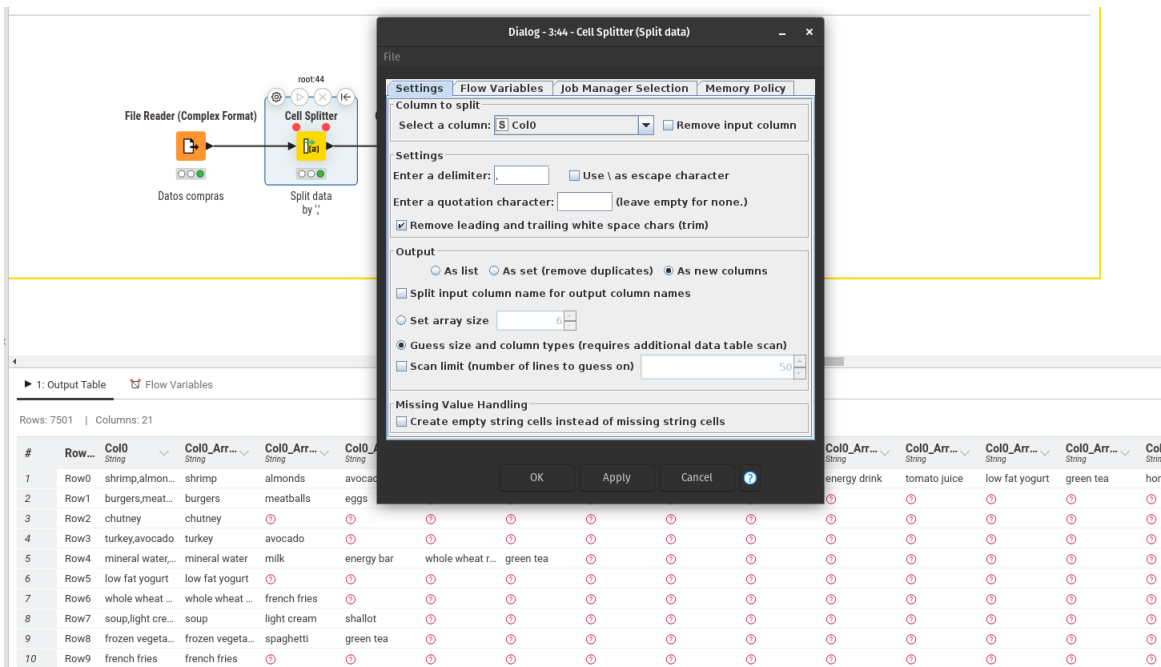


Figura 20: Delimitar los datos por ','.

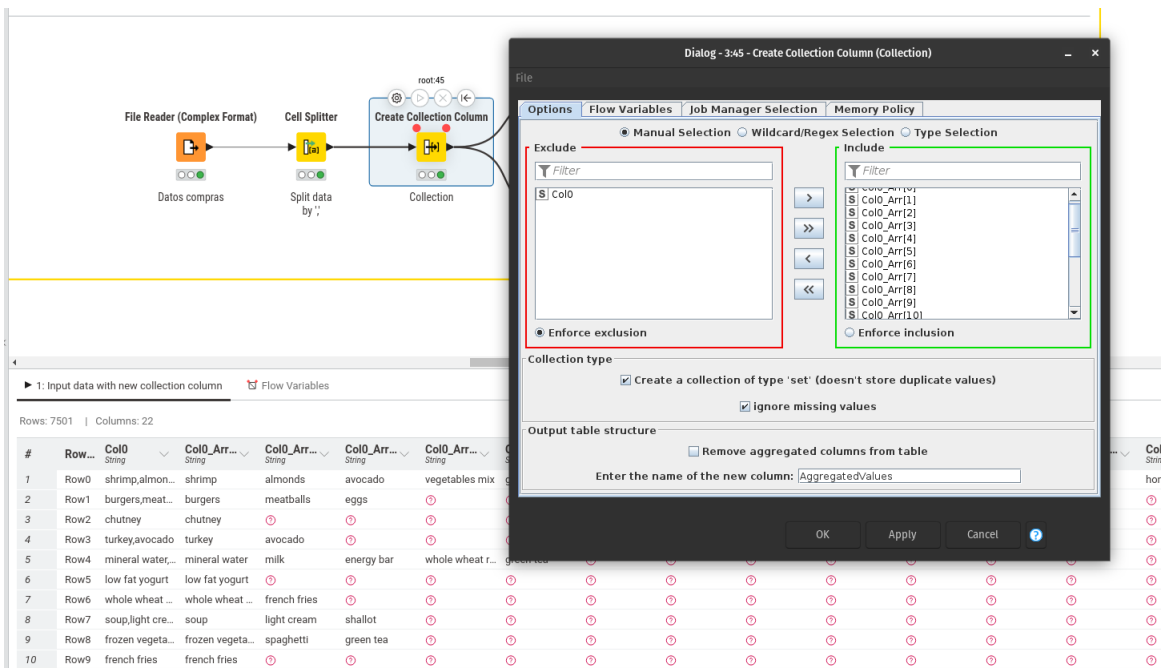


Figura 21: Creación columna elementos comunes con Create Collection Column.

b) Realice un estudio completo para identificar itemsets frecuentes y reglas de asociación importantes que describan correctamente las compras realizadas por los clientes. Para ello ajuste los umbrales de soporte y confianza adecuadamente.

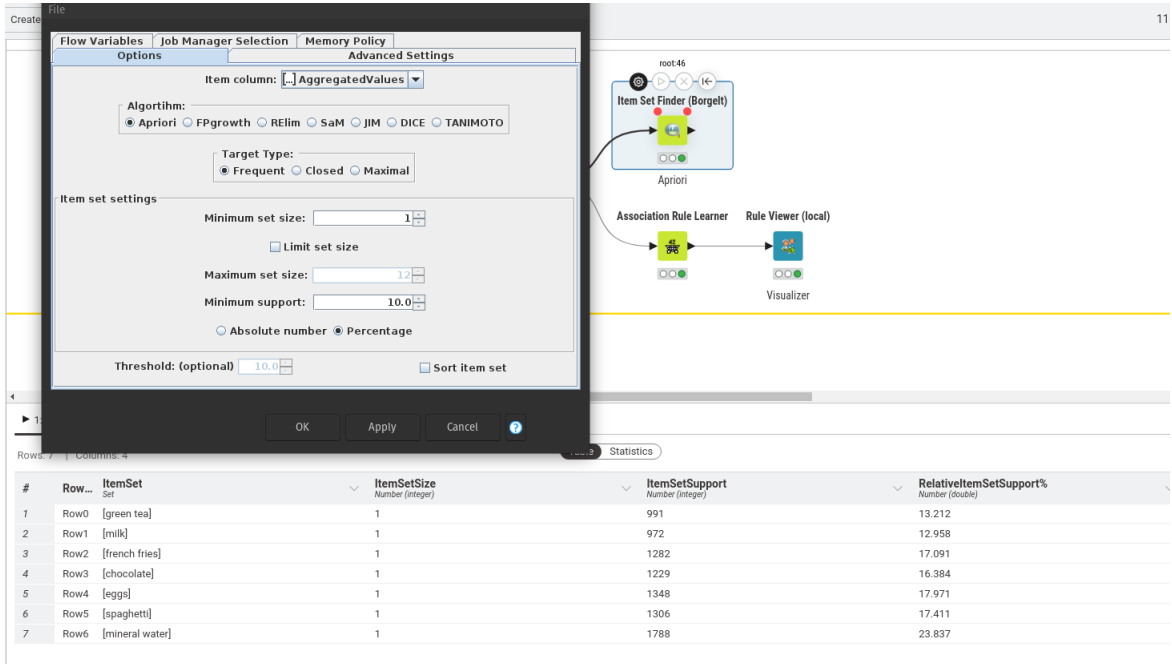


Figura 22: Borgelt de los productos de la compra .

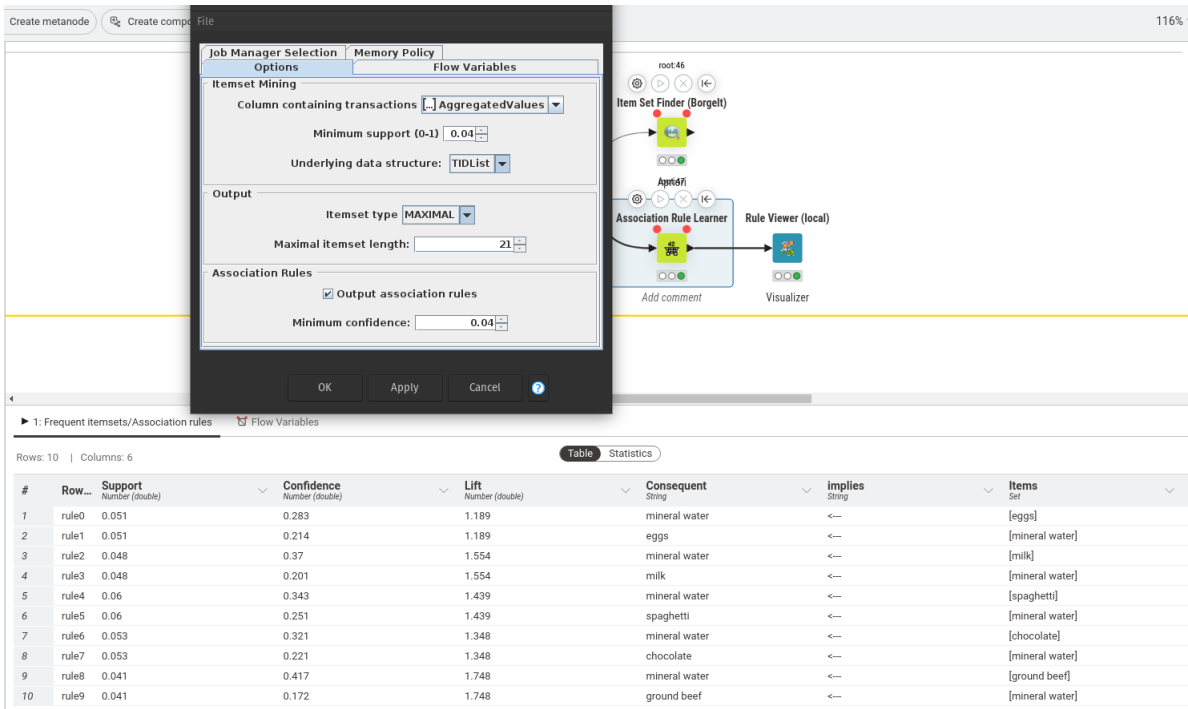


Figura 23: Implicaciones entre elementos de la compra.

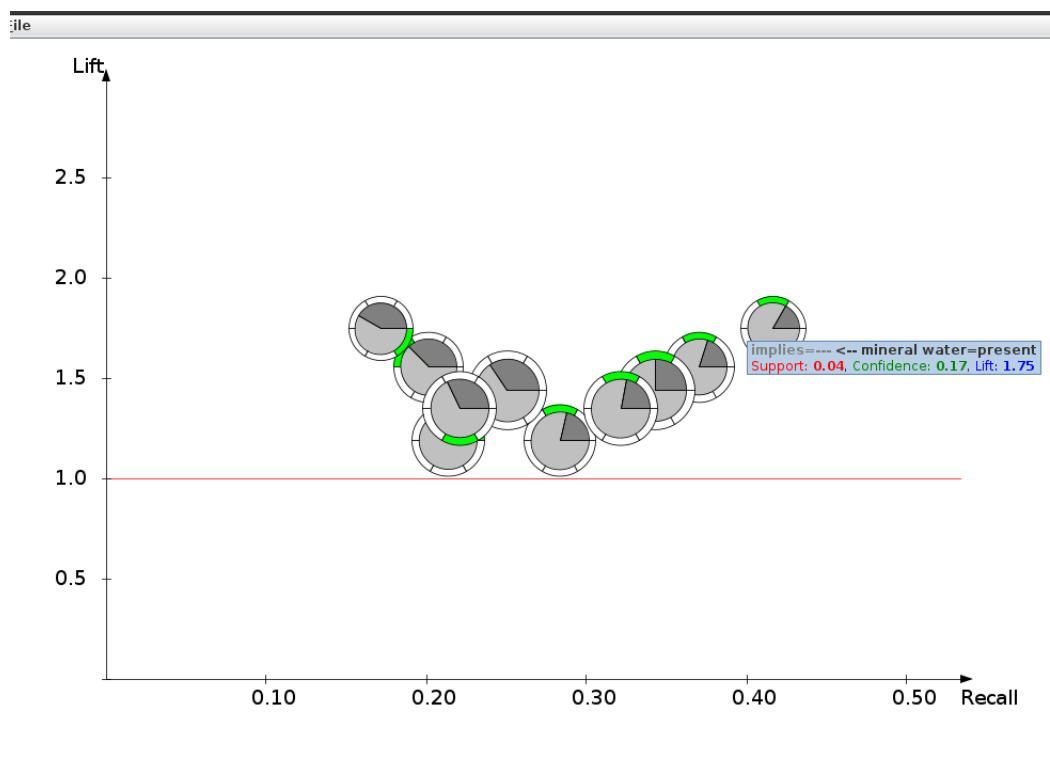


Figura 24: Visualización de los datos filtrados.

Los datos obtenidos una vez se ha bajado el umbral a valores muy bajos (menos de 0.5), se obtienen los siguientes valores:

Datos Filtrados - Item Set:

RowID	ItemSet	ItemSetSize	RelativeItemSetSupport %
1	[green tea]	1	13.212
2	[milk]	1	12.958
3	[french fries]	1	17.091
4	[chocolate]	1	16.384
5	[eggs]	1	17.971
6	[spaghetti]	1	17.411
7	[mineral water]	1	23.837

Cuadro 1: Datos Filtrados - Item Set

RowID	Support	Confidence	Lift	Consequent	implies
1	rule0	0.051	0.283	1.189	mineral water \rightarrow [eggs]
2	rule1	0.051	0.214	1.189	eggs \rightarrow [mineral water]
3	rule2	0.048	0.37	1.554	mineral water \rightarrow [milk]
4	rule3	0.048	0.201	1.554	milk \rightarrow [mineral water]
5	rule4	0.06	0.343	1.439	mineral water \rightarrow [spaghetti]
6	rule5	0.06	0.251	1.439	spaghetti \rightarrow [mineral water]
7	rule6	0.053	0.321	1.348	mineral water \rightarrow [chocolate]
8	rule7	0.053	0.221	1.348	chocolate \rightarrow [mineral water]
9	rule8	0.041	0.417	1.748	mineral water \rightarrow [ground beef]
10	rule9	0.041	0.172	1.748	ground beef \rightarrow [mineral water]

Cuadro 2: Reglas de Asociación Aprendidas

- c) Con los resultados obtenidos, ¿qué estrategias de marketing podría realizar el supermercado para incrementar ventas? ¿Qué sugerencias se le pueden dar a ciertos clientes?

Artículos Frecuentes:

- **Mineral Water:** Este producto aparece con mayor frecuencia en las transacciones, con un soporte de aproximadamente el 24
- **Eggs, Milk, Spaghetti, Chocolate, Ground Beef:** Estos son otros productos comunes con soportes significativos. Estrategias enfocadas en la promoción y colocación estratégica de estos productos podrían aumentar su venta.

Reglas de Asociación Relevantes:

- Mineral Water y Eggs:** Con una confianza del 28
- Mineral Water y Milk:** La relación entre mineral water y milk sugiere una oportunidad para estrategias de marketing conjunto, ya que comprar uno aumenta la probabilidad de comprar el otro.
- Mineral Water y Chocolate:** Aunque el soporte es menor, la alta confianza sugiere que los clientes que compren mineral water también tienden a comprar chocolate. Estrategias que emparejen estos productos podrían ser efectivas.
- Mineral Water y Ground Beef:** Aunque menos común, la relación entre mineral water y ground beef podría indicar oportunidades para promociones o paquetes especiales.

Estrategias de Marketing:

- **Paquetes de Descuentos:** Ofrecer descuentos especiales o paquetes que incluyan mineral water, eggs y milk podría motivar a los clientes a comprar estos productos en conjunto.

-
- **Publicidad Conjunta:** Crear anuncios y campañas promocionales que destaquen las combinaciones de productos más frecuentes, como mineral water con eggs o milk.
 - **Descuentos Temporales:** Implementar descuentos temporales en los productos más frecuentes durante ciertos períodos del día o días de la semana podría estimular la demanda.
 - **Colocación Estratégica en la Tienda:** Colocar productos asociados en ubicaciones cercanas en la tienda puede aumentar la visibilidad y la probabilidad de compra conjunta.

Sugerencias para Clientes:

- **Recomendaciones Personalizadas:** Implementar sistemas de recomendación en la plataforma en línea o en la aplicación móvil del supermercado para sugerir productos adicionales según el historial de compras del cliente.
- **Programas de Recompensas:** Ofrecer recompensas o descuentos adicionales a clientes que participen activamente en la compra de productos asociados.

En resumen, entender las relaciones entre los productos puede ayudar al supermercado a diseñar estrategias de marketing más efectivas y mejorar la experiencia del cliente.

- d) Estudia la posibilidad de obtener reglas negativas del estilo $\text{No item1} \rightarrow \text{No item2}$, ó $\text{item1} \rightarrow \text{No item2}$, ó $\text{No item1} \rightarrow \text{item2}$.

Para este apartado se ha optado por hacerlo en dos vías:

- Crear una columna para cada uno de los elementos que existan en el csv.
 - Usar el nodo Column expressions para ir creando una columna nueva a partir de leer la primera columna, que contiene todos los elementos de cada fila, la Col0.
 - Dentro de este nodo se establece un if/else siendo que si contiene el nombre de un producto, se le asigna el nombre de ese producto, en caso contrario, se le asigna el valor 'No + nombreProducto':

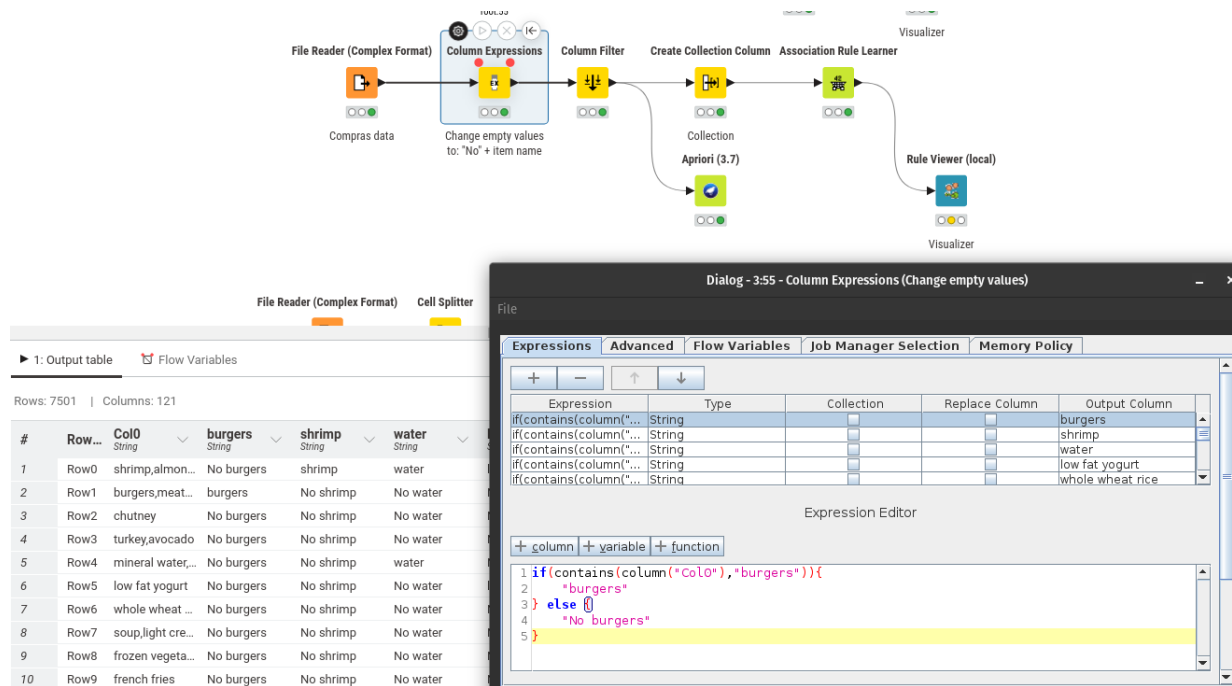


Figura 25: If/else para crear columnas.

- Para determinar todos los productos que hay se ha ido descartando los productos con visual studio y una extensión de csv para poder ver todos los elementos únicos que tiene el documento, de este modo, se ha ido creando un if/else para todos los productos.
- Una vez hecho esto. Se ha filtrado con Column filter ya que son muchos elementos y se queda sin memoria el programa para hacer el cálculo con todas las columnas posibles, de modo que se usan unos cuantos productos como estudio.
- Si usamos el nodo Apriori muestra lo siguiente:

Weka Node View - 3:59 - Apriori (3.7)

File

Apriori

Minimum support: 0.95 (7126 instances)
 Minimum metric <confidence>: 0.9
 Number of cycles performed: 1

Generated sets of large itemsets:

Size of set of large itemsets L(1): 21
 Size of set of large itemsets L(2): 157
 Size of set of large itemsets L(3): 545
 Size of set of large itemsets L(4): 878
 Size of set of large itemsets L(5): 664
 Size of set of large itemsets L(6): 252
 Size of set of large itemsets L(7): 28

Best rules found:

- cottagecheese=Nocottagecheese blacktea=Noblacktea 7163 ==> bramble=Nobramble 7153 <conf:(1)> lift:(1) lev:(0) [3] conv:(1.22)
- cottagecheese=Nocottagecheese 7262 ==> bramble=Nobramble 7251 <conf:(1)> lift:(1) lev:(0) [2] conv:(1.13)
- cottagecheese=Nocottagecheese ketchup=Noketchup 7230 ==> bramble=Nobramble 7219 <conf:(1)> lift:(1) lev:(0) [2] conv:(1.12)
- cottagecheese=Nocottagecheese salad=Nosalad 7228 ==> bramble=Nobramble 7217 <conf:(1)> lift:(1) lev:(0) [2] conv:(1.12)
- cottagecheese=Nocottagecheese strongcheese=Nostrongcheese 7208 ==> bramble=Nobramble 7197 <conf:(1)> lift:(1) lev:(0) [2] conv:(1.12)
- greengrapes=Nogreengrapes cottagecheese=Nocottagecheese 7199 ==> bramble=Nobramble 7188 <conf:(1)> lift:(1) lev:(0) [2] conv:(1.12)
- wholewheatflour=Nowholewheatflour cottagecheese=Nocottagecheese 7199 ==> bramble=Nobramble 7188 <conf:(1)> lift:(1) lev:(0) [2] conv:(1.12)
- cottagecheese=Nocottagecheese antioxydantjuice=Nantioxydantjuice 7197 ==> bramble=Nobramble 7186 <conf:(1)> lift:(1) lev:(0) [2] conv:(1.12)
- cottagecheese=Nocottagecheese ketchup=Noketchup 7196 ==> bramble=Nobramble 7185 <conf:(1)> lift:(1) lev:(0) [2] conv:(1.12)
- cottagecheese=Nocottagecheese magazines=Nomagazines 7187 ==> bramble=Nobramble 7176 <conf:(1)> lift:(1) lev:(0) [2] conv:(1.12)

Figura 26: Nodo Apriori con algunos datos de la compra.

-
- a) cottagecheese=Nocottagecheese blacktea=Noblacktea
7163 ==>bramble=Nobramble 7153
<conf:(1)>lift:(1)
lev:(0) [3] conv:(1.22)
 - b) cottagecheese=Nocottagecheese 7262 ==>
bramble=Nobramble 7251 <conf:(1)>lift:(1)
lev:(0) [2] conv:(1.13)
 - c) cottagecheese=Nocottagecheese ketchup=Noketchup
7230 ==>bramble=Nobramble 7219
<conf:(1)>lift:(1)
lev:(0) [2] conv:(1.12)
 - d) cottagecheese=Nocottagecheese salad=Nosalad
7228 ==>bramble=Nobramble 7217
<conf:(1)>lift:(1)
lev:(0) [2] conv:(1.12)
 - e) cottagecheese=Nocottagecheese strongcheese=Nostrongcheese
7208 ==>bramble=Nobramble 7197
<conf:(1)>lift:(1)
lev:(0) [2] conv:(1.12)
 - f) greengrapes=Nogreengrapes cottagecheese=Nocottagecheese
7199 ==>bramble=Nobramble 7188
<conf:(1)>lift:(1)
lev:(0) [2] conv:(1.12)
 - g) wholewheatflour=Nowholewheatflour cottagecheese=Nocottagecheese
7199 ==>bramble=Nobramble 7188
<conf:(1)>lift:(1)
lev:(0) [2] conv:(1.12)
 - h) cottagecheese=Nocottagecheese antioxydantjuice=Noantioxydantjuice
7197 ==>bramble=Nobramble 7186
<conf:(1)>lift:(1)
lev:(0) [2] conv:(1.12)
 - i) cottagecheese=Nocottagecheese salad=Nosalad ketchup=Noketchup
7196 ==>bramble=Nobramble 7185
<conf:(1)>lift:(1)
lev:(0) [2] conv:(1.12)
 - j) cottagecheese=Nocottagecheese magazines=Nomagazines
7187 ==>bramble=Nobramble 7176
<conf:(1)>lift:(1)
lev:(0) [2] conv:(1.12)

- Finalmente, si usamos el nodo Association Rule Learner se pueden ver asociaciones de los tres tipos que se pide:

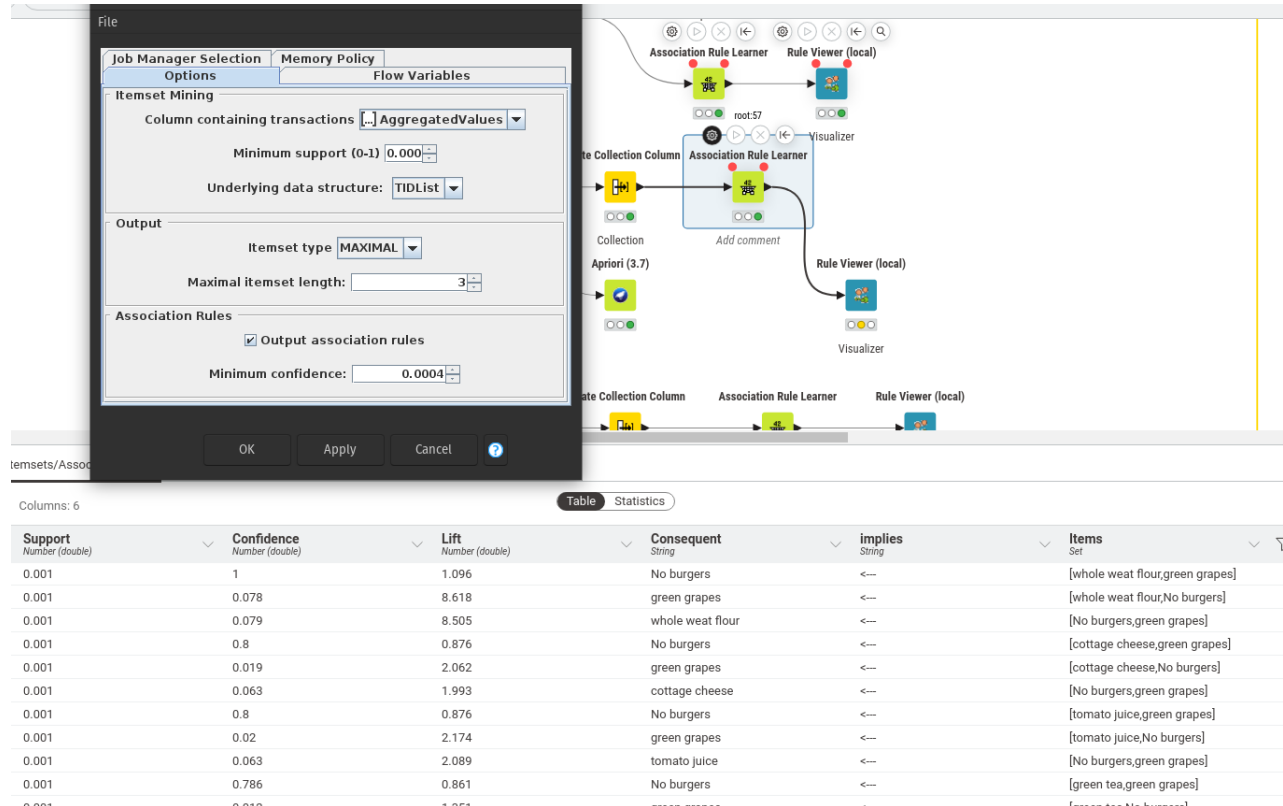


Figura 27: Nodo Association Rule Learner con algunos datos de la compra.

Consequent String	implies String	Items Set
No burgers	<---	[whole wheat flour,green grapes]
green grapes	<---	[whole wheat flour,No burgers]
whole wheat flour	<---	[No burgers,green grapes]
No burgers	<---	[cottage cheese,green grapes]
green grapes	<---	[cottage cheese,No burgers]
cottage cheese	<---	[No burgers,green grapes]
No burgers	<---	[tomato juice,green grapes]
green grapes	<---	[tomato juice,No burgers]
tomato juice	<---	[No burgers,green grapes]
No burgers	<---	[green tea,green grapes]

Figura 28: Nodo Association Rule Learner ejemplos.

- Por ejemplo, se puede ver que 'Green Grapes y Flour wheat whole' implica no burgers, 'No burgers y Cottage cheese' implica 'Green Grapes'. Hay que destacar que casi todos los valores son del tipo 'No más nombre del producto' por lo que se ha tenido que reducir el umbral de confianza y soporte a muy bajo para mostrar asociaciones con valores de tipo que si tienen con tipo no tienen.

- Si visualizamos los datos se puede ver que son muy dispersos y se concentran casi todos los datos cuando la tasa de soporte y de confianza es superior a 0.7 por lo mencionado anteriormente, que casi todos los datos son de tipo 'No más nombre producto':

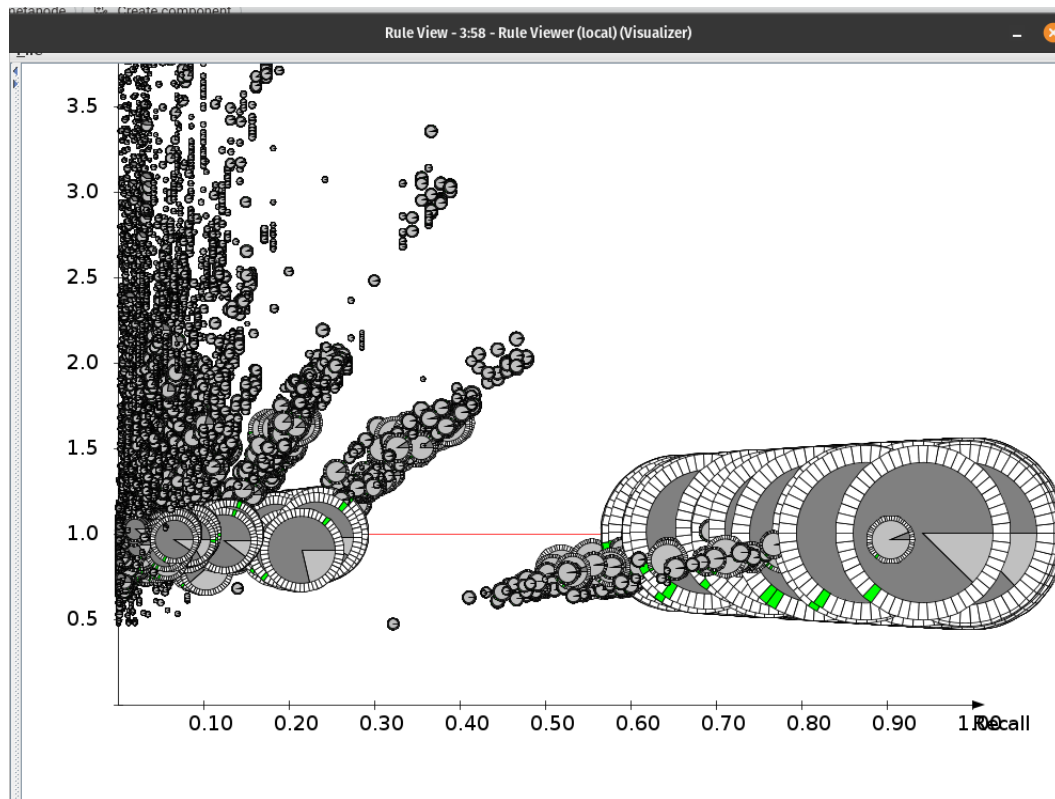


Figura 29: Representación de la dispersión de los datos.

- Rellenar los datos del caso anterior de nulo a una cadena de caracteres que indique el nombre de la tabla.

En este caso, se han ido cambiando los datos vacíos con el nodo 'Missing Values' a 'No_item_X' donde 'X' es el número de la columna, en este caso solo tendría esa asignación si es vacío, pero para este caso, se pueden usar todas columnas que tiene cuando se ha usado previamente el nodo cell splitter.

- Dando como resultado esto:

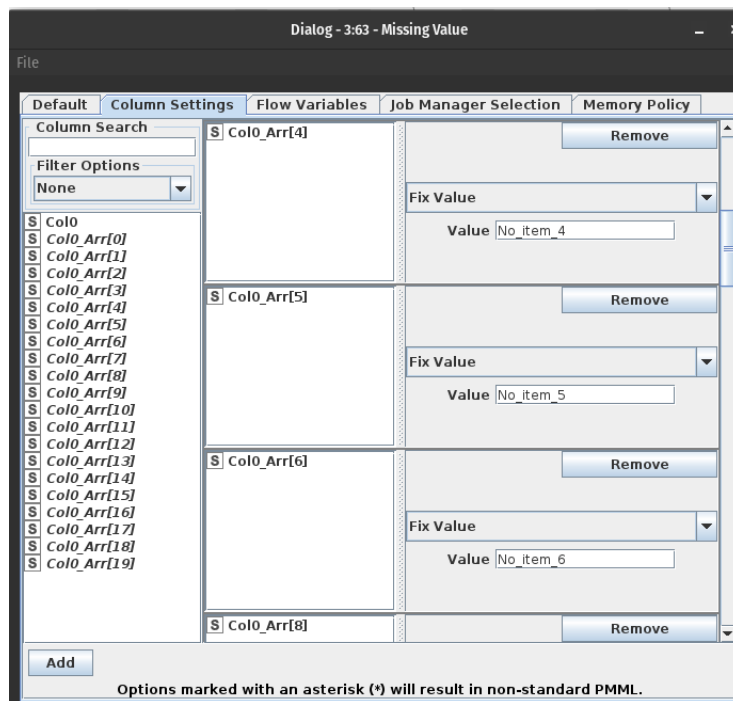


Figura 30: Configuración del nodo Missing Value para No.Item.X.

Col0_Arr... String	Col0_Arr... String	Col0_Arr... String	Col0_Arr... String	Col0_Arr... String	Col0_Arr... String	Col0_Arr... String	Col0_Arr... String	Col0_Arr... String
shrimp	almonds	avocado	vegetables mix	green grapes	whole wheat fl...	yams	cottage cheese	energy drink
burgers	meatballs	eggs	No_item_3	No_item_4	No_item_5	No_item_6	No_item_7	No_item_8
chutney	No_item_1	No_item_2	No_item_3	No_item_4	No_item_5	No_item_6	No_item_7	No_item_8
turkey	avocado	No_item_2	No_item_3	No_item_4	No_item_5	No_item_6	No_item_7	No_item_8
mineral water	milk	energy bar	whole wheat r...	green tea	No_item_5	No_item_6	No_item_7	No_item_8
low fat yogurt	No_item_1	No_item_2	No_item_3	No_item_4	No_item_5	No_item_6	No_item_7	No_item_8
whole wheat ...	french fries	No_item_2	No_item_3	No_item_4	No_item_5	No_item_6	No_item_7	No_item_8
soup	light cream	shallot	No_item_3	No_item_4	No_item_5	No_item_6	No_item_7	No_item_8
frozen vegeta...	spaghetti	green tea	No_item_3	No_item_4	No_item_5	No_item_6	No_item_7	No_item_8
french fries	No_item_1	No_item_2	No_item_3	No_item_4	No_item_5	No_item_6	No_item_7	No_item_8

Figura 31: Resultado del uso del nodo Missing Value para No.Item.X.

- Una vez hecho esto, podemos usar el nodo Association Rule Learner del mismo modo que antes, utilizando umbrales muy bajos y tiene asociaciones de los 3 tipos que se piden, pero no da información de qué producto es.

Confidence Number (double)	Lift Number (double)	Consequent String	implies String	Items Set
0.12	0.905	green tea	←	[No_item_9,No_item_8]
0.924	1.013	No_item_8	←	[No_item_9,green tea]
1	1.056	No_item_9	←	[No_item_8,green tea]
0.12	0.905	green tea	←	[No_item_8,No_item_10]
0.893	0.978	No_item_8	←	[No_item_10,green tea]
1	1.035	No_item_10	←	[No_item_8,green tea]
0.12	0.905	green tea	←	[No_item_11,No_item_8]
0.866	0.948	No_item_8	←	[No_item_11,green tea]
1	1.021	No_item_11	←	[No_item_8,green tea]
0.12	0.905	green tea	←	[No_item_8,No_item_12]
0.851	0.993	No_item_8	←	[No_item_12,green tea]

Figura 32: Implicaciones entre productos en el caso No.Item.X.

- Por último, visualizamos los datos para ver la distribución de los datos y, como en el caso anterior, hay mayor concentración tanto en valores inferiores 0.3 y en valores superiores a 0.7

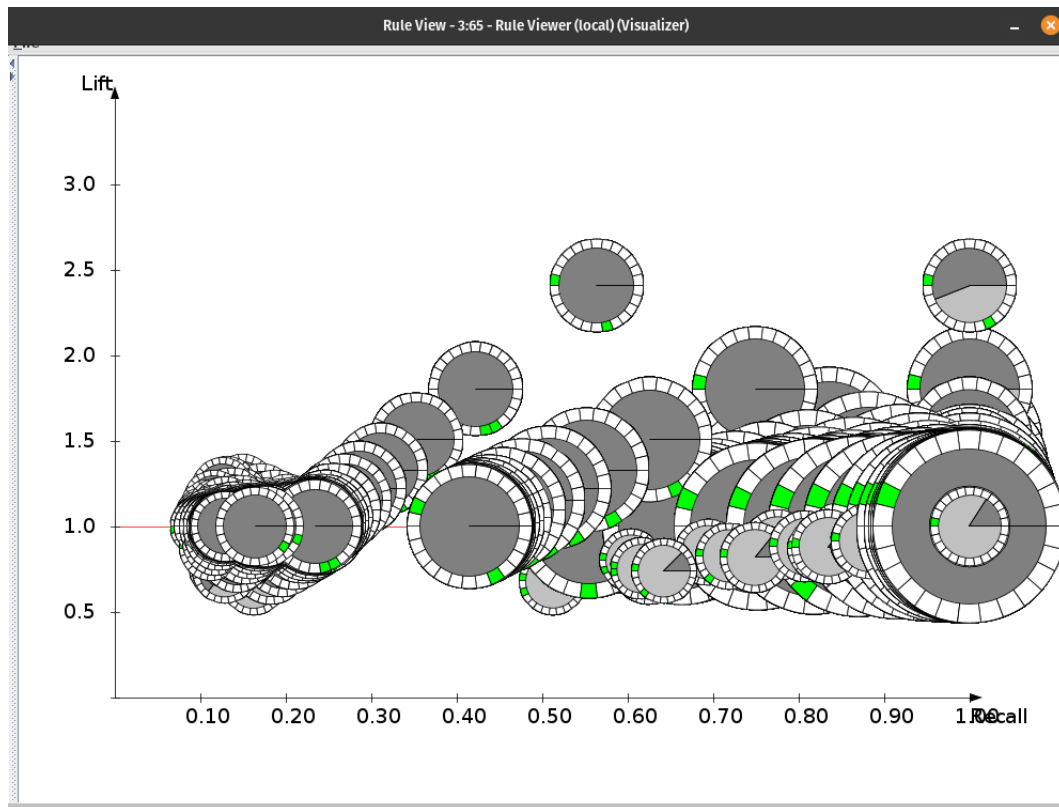


Figura 33: Representación de la dispersión en el caso No_Item_X.