UNIVERSIDAD DE GRANADA

Reglas de Asociación con Knime

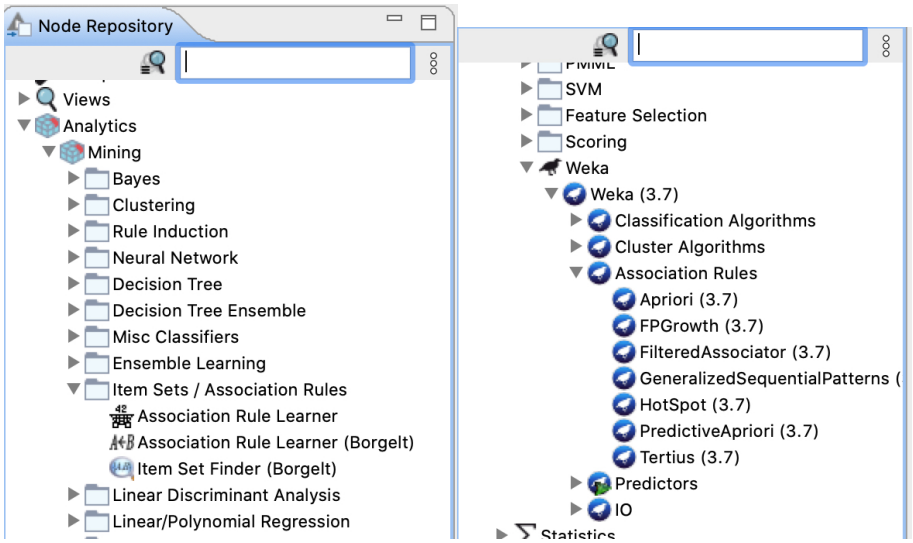
Tratamiento Inteligente de Datos
Master Universitario en Ingeniería Informática

1. Objetivos y evaluación

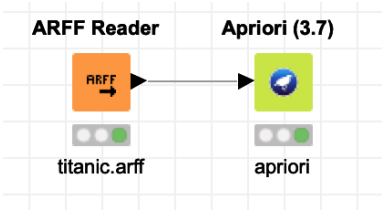
En esta práctica veremos el uso de algoritmos de generación de reglas de asociación en Knime. Se trabajará con un conjunto de datos reales sobre el que se emplearán diferentes algoritmos de obtención de reglas de asociación (para su comparación) y a la luz del conocimiento descubierto se podrán concluir estrategias para resolver el problema.

2. Reglas de asociación

Knime cuenta con varios nodos para el cálculo de reglas de asociación en la carpeta `Analytics/Mining/Item Sets / Association Rules` y en la carpeta de nodos de Weka `Analytics/Mining/Weka/Weka (3.7)/Association Rules`.



Vamos a considerar un workflow con el siguiente aspecto



Donde hemos utilizado un nodo de lectura de ficheros `arff`. A través de dicho nodo cargaremos los datos del fichero `titanic.arff`. El fichero `titanic.arff` contiene datos sobre las características de los 2201 pasajeros del Titanic. Estos datos son reales y provienen del *Report on the Loss of the "Titanic"(S.S.)* (British Board of Trade , Inquiry Report (reprint), Gloucester, UK, Allan Sutton Publishing, 1990). El formato `arff` (Attribute-Relation File Format) consiste, simplemente, en un fichero de texto en el que se almacena una tabla de datos, con una línea por tupla y los valores de una misma tupla separados por comas (en la misma línea del fichero de

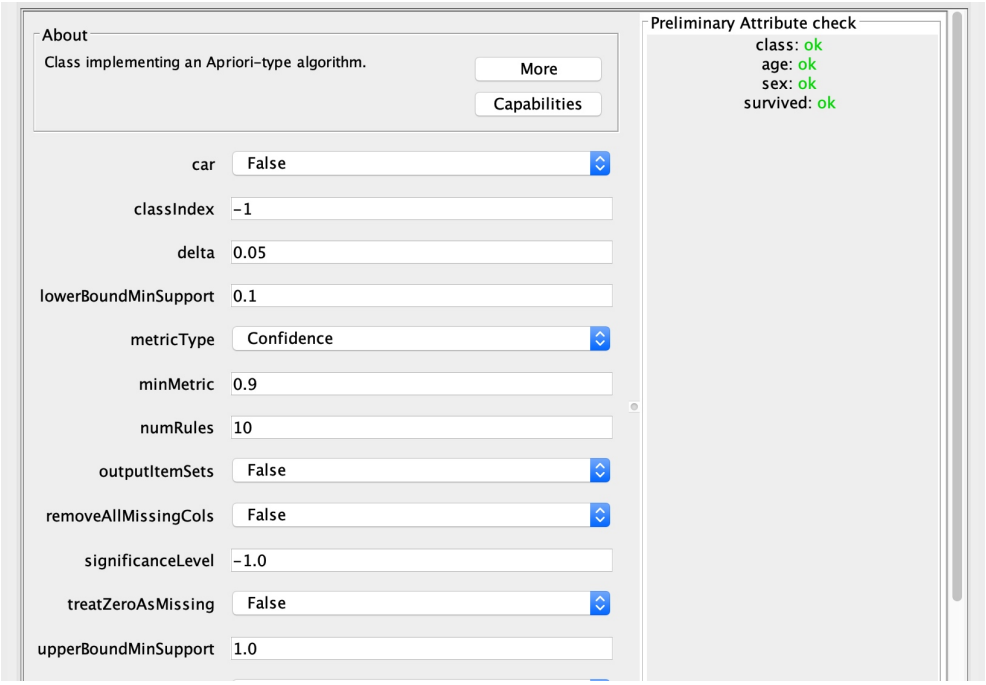
texto). Adicionalmente, los ficheros `arff` incluyen una cabecera con información adicional acerca de los nombres y tipos de datos asociados a los distintos atributos atributos de la relación, tal como se muestra a continuación

```
% Comentarios

@RELATION Persona

@ATTRIBUTE Ingresos NUMERIC
@ATTRIBUTE Nombre string
@ATTRIBUTE FechaNacimiento date
@ATTRIBUTE CategoriaLaboral {Administrativo, Seguridad, Directivo}

@DATA

18000.34 , Juan , 1979
22300.05 , Inma, 1967
........
```

En el caso del fichero de datos de los tripulantes del Titanic, sólo consideraremos los siguientes cuatro atributos, que ya aparecen codificados en el fichero `titanic.arff`

- class (crew, 1st, 2nd, 3rd)
- age (adult, child)
- sex (male, female)
- survived (yes, no)

El nodo `Apriori (3.7)` de Weka ejecuta el algoritmo Apriori para la obtención de reglas sobre un conjunto de datos. A pesar de que la base de datos no es transaccional, este nodo admite bases de datos relacionales considerando cada valor del atributo como un item del tipo `atributo = valor`. Por ejemplo, en el archivo `titanic.arff` considera dos items del atributo `age`: `age = adult` y `age = child`. Este nodo sólo admite datos nominales, por lo que los numéricos deben ser discretizados (por ejemplo).



En la configuración del nodo es posible determinar el soporte mínimo, la métrica y el valor mínimo de esta para las reglas a obtener (confianza, lift, leverage, conviction). `OutputItemsetItems` indica si deseamos mostrar los itemsets frecuentes.. Una explicación de cada opción viene dada al pulsar el botón `More`. En este caso fijamos

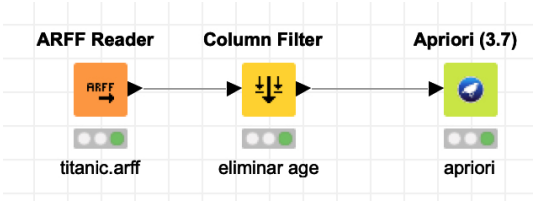
- Un umbral de soporte mínimo del 10 % (0.1 representa un 10 % en `lowerBoundMinSuppport`)
- Confianza del 90 % (0.9 en `minMetric`).
- Que muestre las 10 mejores reglas (`numRules`).
- Que no muestre los itemsets frecuentes.

Al visualizar los resultados del nodo `Apriori (3.7)` obtenemos las reglas generadas

File

```
Apriori
=======

Minimum support: 0.35 (770 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 13

Generated sets of large itemsets:

Size of set of large itemsets L(1): 4

Size of set of large itemsets L(2): 5

Size of set of large itemsets L(3): 2

Best rules found:

 1. class=crew 885 ==> age=adult 885    <conf:(1)> lift:(1.05) lev:(0.02) [43] conv:(43.83)
 2. class=crew sex=male 862 ==> age=adult 862    <conf:(1)> lift:(1.05) lev:(0.02) [42] conv:(42.69)
 3. sex=male survived=no 1364 ==> age=adult 1329    <conf:(0.97)> lift:(1.03) lev:(0.01) [32] conv:(1.88)
 4. class=crew 885 ==> sex=male 862    <conf:(0.97)> lift:(1.24) lev:(0.08) [165] conv:(7.87)
 5. class=crew age=adult 885 ==> sex=male 862    <conf:(0.97)> lift:(1.24) lev:(0.08) [165] conv:(7.87)
 6. class=crew 885 ==> age=adult sex=male 862    <conf:(0.97)> lift:(1.29) lev:(0.09) [191] conv:(8.95)
 7. survived=no 1490 ==> age=adult 1438    <conf:(0.97)> lift:(1.02) lev:(0.01) [21] conv:(1.39)
 8. sex=male 1731 ==> age=adult 1667    <conf:(0.96)> lift:(1.01) lev:(0.01) [21] conv:(1.32)
 9. age=adult survived=no 1438 ==> sex=male 1329    <conf:(0.92)> lift:(1.18) lev:(0.09) [198] conv:(2.79)
10. survived=no 1490 ==> sex=male 1364    <conf:(0.92)> lift:(1.16) lev:(0.09) [192] conv:(2.51)
```

Si añadimos el nodo `Analytics/Statistics/Statistics` comprobaremos que, para el atributo `age`, hay 2092 tuplas con `adult` y sólo 109 con valor `child`. En clase de teoría hemos comentado los problemas que surgen con la presencia de ítems demasiado frecuentes. Para eliminar su influencia en nuestro análisis, podemos añadir un nodo `Column Filter` para eliminar la columna `age`.



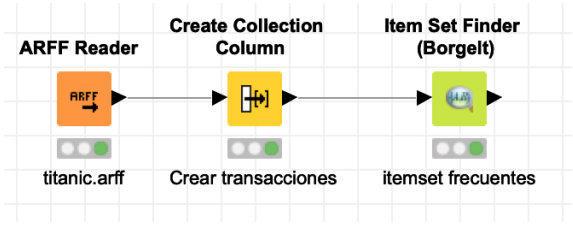
Si hace lo indicado en la opción anterior, obviamente, no se generará ninguna regla relativa a `age = child`.

File

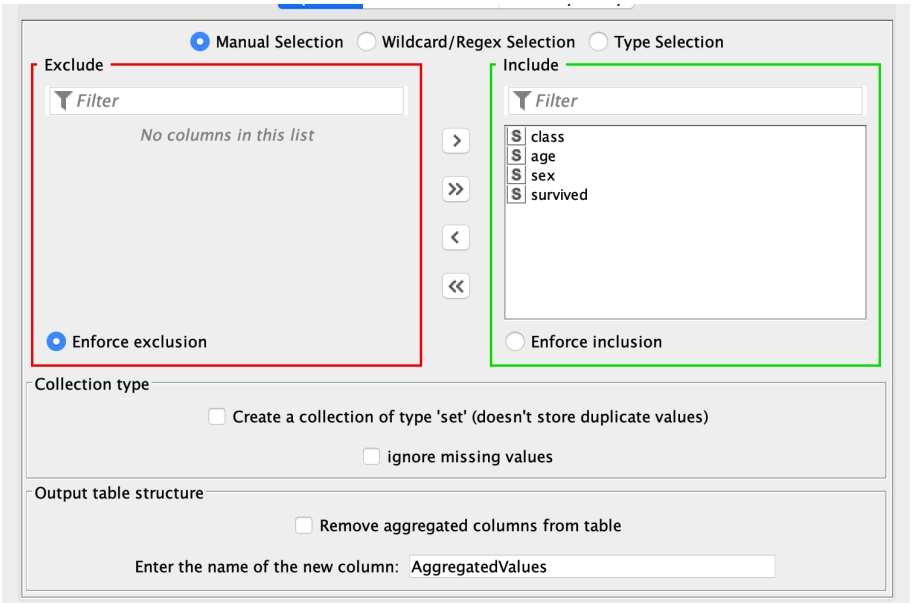
```
Apriori
=======

Minimum support: 0.1 (220 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 18

Generated sets of large itemsets:

Size of set of large itemsets L(1): 8

Size of set of large itemsets L(2): 7

Size of set of large itemsets L(3): 2

Best rules found:

 1. class=crew survived=no 673 ==> sex=male 670    <conf:(1)> lift:(1.27) lev:(0.06) [140] conv:(35.93)
 2. class=crew 885 ==> sex=male 862    <conf:(0.97)> lift:(1.24) lev:(0.08) [165] conv:(7.87)
 3. survived=no 1490 ==> sex=male 1364    <conf:(0.92)> lift:(1.16) lev:(0.09) [192] conv:(2.51)
```

Lo ideal, no obstante, sería generar las reglas utilizando todos los datos disponibles y, posteriormente, filtrar las reglas obtenidas. Desgraciadamente, ni KNIME ni Weka proporcionan herramientas para hacerlo.

Los nodos contenidos en `Analytics/Mining/Item Sets / Association Rules` sí necesitan actuar sobre una base de datos trasaccional. Para transformar la base de datos relacional de `titanic.arff`, utilizamos el nodo `Create Collection Column` en la carpeta `Manipulation/Column/Split & Combine`.

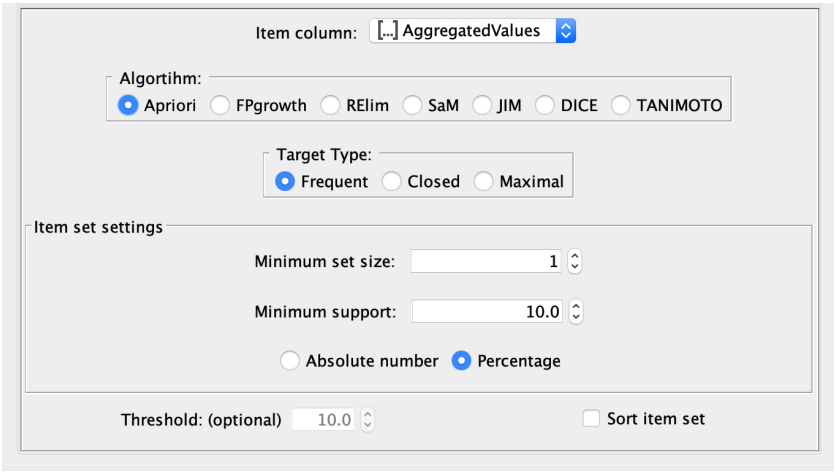


Este nodo crea una nueva columna donde agrupa el valor de los atributos seleccionados de una instancia en un conjunto.



Row ID	class	age	sex	survived	AggregatedValues
Row0	1st	adult	male	yes	[1st,adult,male,...]
Row1	1st	adult	male	yes	[1st,adult,male,...]
Row2	1st	adult	male	yes	[1st,adult,male,...]
Row3	1st	adult	male	yes	[1st,adult,male,...]
Row4	1st	adult	male	yes	[1st,adult,male,...]
Row5	1st	adult	male	yes	[1st,adult,male,...]
Row6	1st	adult	male	yes	[1st,adult,male,...]
Row7	1st	adult	male	yes	[1st,adult,male,...]
Row8	1st	adult	male	yes	[1st,adult,male,...]
Row9	1st	adult	male	yes	[1st,adult,male,...]
Row10	1st	adult	male	yes	[1st,adult,male,...]

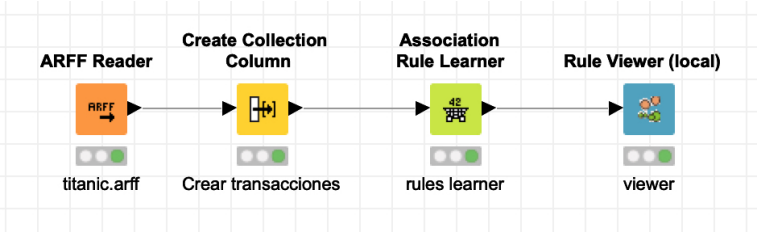
Ahora sí podemos configurar el nodo `Item Set Finder (Borgelt)` para encontrar itemsets frecuentes.

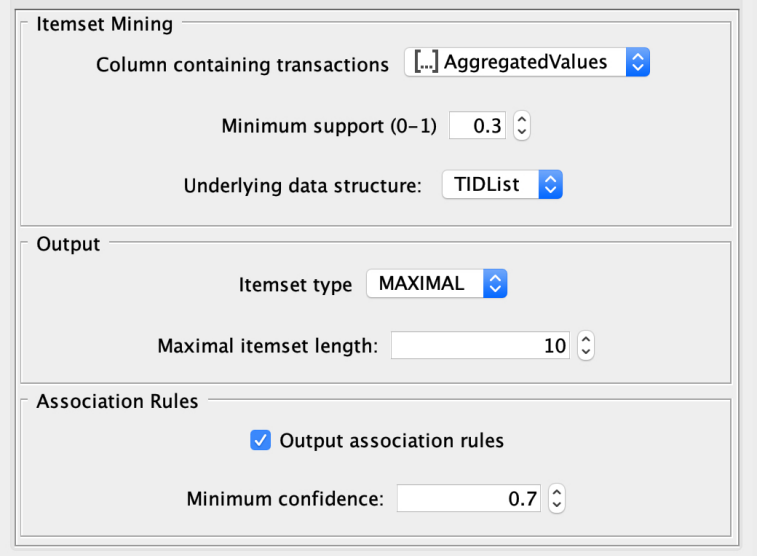


Que ofrece como resultado

Row ID	ItemSet	ItemSetSize	ItemSetSupport	RelativeItemSetSupport%
Row0	[2nd]	1	285	12.949
Row1	[2nd,adult]	2	261	11.858
Row2	[1st]	1	325	14.766
Row3	[1st,adult]	2	319	14.493
Row4	[female]	1	470	21.354
Row5	[female,yes]	2	344	15.629
Row6	[female,ye...	3	316	14.357
Row7	[female,a...	2	425	19.309
Row8	[3rd]	1	706	32.076
Row9	[3rd,no]	2	528	23.989
Row10	[3rd,no,m...	3	422	19.173
Row11	[3rd,no,m...	4	387	17.583
Row12	[3rd,no,a...	3	476	21.627
Row13	[3rd,male]	2	510	23.171
Row14	[3rd,male...	3	462	20.99
Row15	[3rd,adult]	2	627	28.487
Row16	[yes]	1	711	32.303
Row17	[yes,male]	2	367	16.674

Para calcular las reglas de asociación podemos utilizar el nodo `Association Rule Learner`

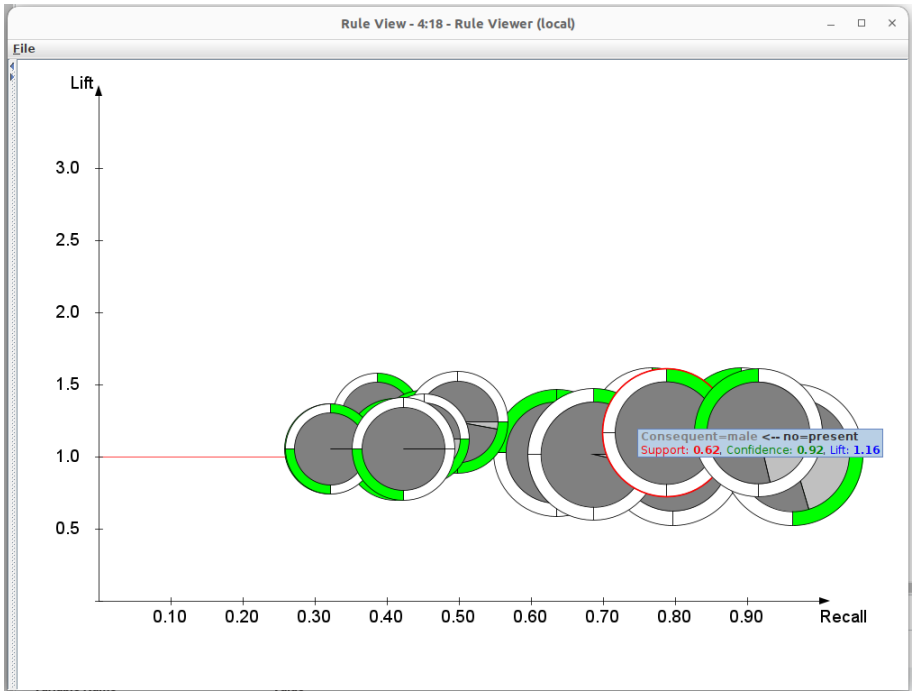




Obteniendo como resultado

Row ID	Support	Confidence	Lift	Consequent	implies	Items
rule0	0.304	1	1.052	adult	<---	[no,male,crew]
rule1	0.304	0.996	1.266	male	<---	[no,adult,crew]
rule2	0.304	0.777	1.148	no	<---	[adult,male,crew]
rule3	0.604	0.974	1.025	adult	<---	[no,male]
rule4	0.604	0.924	1.175	male	<---	[no,adult]
rule5	0.604	0.797	1.178	no	<---	[adult,male]
rule6	0.392	1	1.052	adult	<---	[male,crew]
rule7	0.392	0.974	1.238	male	<---	[adult,crew]
rule8	0.757	0.963	1.013	adult	<---	[male]
rule9	0.757	0.797	1.013	male	<---	[adult]
rule10	0.306	1	1.052	adult	<---	[no,crew]
rule11	0.306	0.76	1.123	no	<---	[adult,crew]
rule12	0.653	0.965	1.015	adult	<---	[no]
rule13	0.402	1	1.052	adult	<---	[crew]
rule14	0.62	0.915	1.164	male	<---	[no]
rule15	0.62	0.788	1.164	no	<---	[male]

Además, las reglas pueden visualizarse con el nodo `Rule Viewer (local)`, donde cada regla es representada por un círculo cuyo tamaño viene dado por el soporte y en el interior hay un diagrama de pastel representando la confidencia de la regla. El borde del círculo se divide en tantos segmentos como número de items diferentes hay en el dataset. Si un item aparece en el antecedente de la regla, su segmento asociado se colorea de verde. Como se habrá visto en clase de teoría, el lift mide la frecuencia de un patrón observado (una regla) con respecto a lo que se observaría por azar, y la recall mide cuántos items relevantes son seleccionados.



3. 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.

(b) La base de datos está en forma relacional, donde cada atributo tiene valor 1 ó 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).

(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`.

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

(e) Discuta las relaciones interesantes que encuentre.

4. 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.

(b) Relice 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 confidencia adecuadamente.

(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?.

(d) Estudia la posibilidad de obtener reglas negativas del estilo `No item1` $\rightarrow$ `No item2`, ó `item1` $\rightarrow$ `No item2`, ó `No item1` $\rightarrow$ `item2`.