

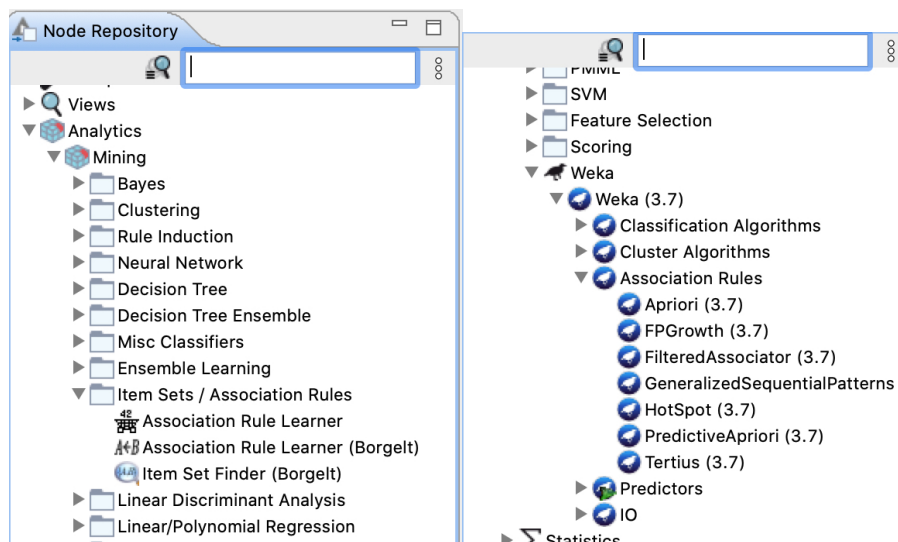


1. Objetivos y evaluación

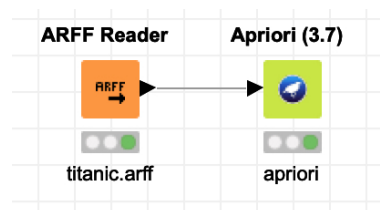
En esta práctica veremos el uso de algoritmos de generación de reglas de asociación en Knime. Se trabajará con un conjunto de datos reales sobre el que se emplearán diferentes algoritmos de obtención de reglas de asociación (para su comparación) y a la luz del conocimiento descubierto se podrán concluir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

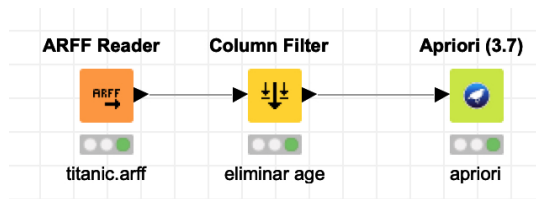
Best rules found:

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)

```

El funcionamiento es similar para los diferentes algoritmos contenidos en los nodos de Weka, excepto para el nodo **FPGrowth** que solo admite atributos binarios. En el caso del fichero `titanic.arff`, el atributo `class` debe ser filtrado.

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:

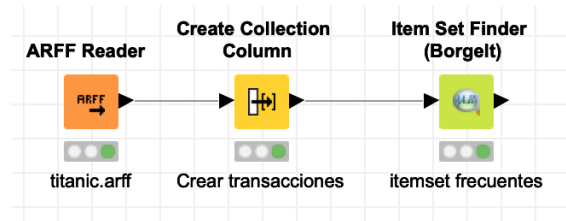
Size of set of large itemsets L(1): 8
Size of set of large itemsets L(2): 7
Size of set of large itemsets L(3): 2

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.

La interfaz de configuración del nodo **Create Collection Column** muestra las siguientes opciones:

- Selection Method**: ☒ Manual Selection, ☐ Wildcard/Regex Selection, ☐ Type Selection.
- Exclude** (resaltado con un recuadro rojo):
 - Filter: (contiene "No columns in this list").
 - ☒ Enforce exclusion.
- Include** (resaltado con un recuadro verde):
 - Filter: (contiene "class", "age", "sex", "survived").
 - ☐ Enforce inclusion.
- Collection type**:
 - ☐ Create a collection of type 'set' (doesn't store duplicate values).
 - ☐ ignore missing values.
- Output table structure**:
 - ☐ Remove aggregated columns from table.
 - Enter the name of the new column: (contiene "AggregatedValues").

Row ID	S class	S age	S sex	S 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,...]
Row11	1st	adult	male	yes	[1st,adult,male,...]

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

Item column: [...] AggregatedValues

Algoritmo:
☒ Apriori
☐ FPGrowth
☐ RELim
☐ SaM
☐ JIM
☐ DICE
☐ TANIMOTO

Target Type:
☒ Frequent
☐ Closed
☐ Maximal

Item set settings

Minimum set size: 1

Minimum support: 10.0

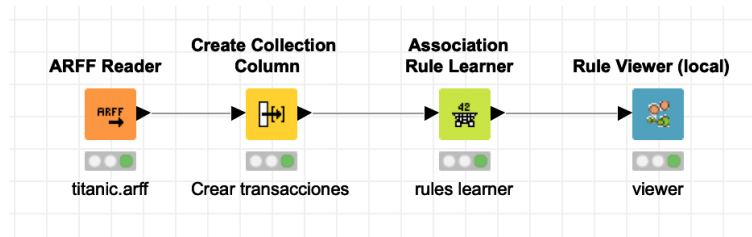
☐ Absolute number
☒ Percentage

Threshold: (optional) 10.0
☐ Sort item set

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
Row18	[yes,male]	2	338	15.357

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



Itemset Mining

Column containing transactions
[... AggregatedValues

Minimum support (0-1)
0.3

Underlying data structure:
TIDList

Output

Itemset type
MAXIMAL

Maximal itemset length:
10

Association Rules

☒ Output association rules

Minimum confidence:
0.7

Obteniendo como resultado

Row ID	D Support	D Confidence	D Lift	S Consequent	S 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]

3. UniversalBank

UniversalBank es un banco relativamente joven que está creciendo rápidamente en términos de captación de clientes. La mayoría de ellos son clientes de pasivo (es decir, clientes con predominio de contratos de productos financieros de ahorro e inversión y, por tanto, aquellos sobre los que el banco tiene una serie de obligaciones de pago) con diferentes tamaños de relación con el banco. La base de clientes de activo (es decir, aquellos con predominio de contratos de productos financieros de financiación y, por tanto, clientes sobre los que el banco tiene una serie de derechos de cobro) es bastante pequeña y el banco quiere crecer esta base rápidamente para traer más negocios de préstamos. En concreto, el banco quiere estudiar la manera de convertir sus clientes de pasivo en clientes de activo.

El año pasado, el banco desarrolló una campaña para clientes de pasivo (ver `prestamo.xls`) con la que se consiguió una conversión del 9%. Esto ha alentado al departamento de marketing para diseñar campañas más inteligentes y con un objetivo mejor definido. En concreto, se persigue responder a la siguiente cuestión: Hay varios productos/servicios que el banco ofrece tales como cuentas de valores (securities accounts), certificados de depósitos, servicios de banca en línea o tarjetas de crédito. ¿Se puede detectar alguna relación entre estos productos para encontrar oportunidades de venta cruzada? Para responder a la pregunta se propone obtener reglas de asociación mediante alguno de los algoritmos de generación de reglas.

Debe aplicarse un preprocesamiento de datos ya que existen atributos que no son nominales, o que no parecen interesantes para este problema. También es posible que filtrar algunos atributos den lugar a reglas de asociación interesantes.