



UNIVERSIDAD DE GRANADA

TRATAMIENTO INTELIGENTE DE DATOS

Práctica 3

Autor

Antonio José Muriel Gálvez



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE TELECOMUNICACIÓN

Granada, 20 de noviembre de 2024

Índice

Ejercicio 1	2
Enunciado	2
Solución Obtenida en Knime	2
Nodo File Reader	3
Nodo Number to String	3
Nodo Apriori 3.7	3
Nodo Metanodo 1	4
Nodo Metanodo 2	6
Nodo Apriori 3.7	6
Nodo Create Collection Column	8
Nodo Item Set Finder (Borgelt)	8
Nodo Association Rule Learner	9
 Ejercicio 2	 12
Enunciado	12
Solución KNIME	13
Nodo File Reader (Complex Format	13
Nodo Cell Splitter	14
Nodo Create Collection Column	14
Nodo Item Set Finder (Borgelt)	15
Nodo Association Rule Learner (Borgelt)	16
Identificación de Itemsets Frecuentes	17
Generación de Reglas de Asociación	17
Exploración de Reglas Negativas	17
Propuestas de Estrategias de Marketing	18

Ejercicio 1

Enunciado

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.

- Parte (a): Lee la base de datos en formato CSV.
- Parte (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, puedes aplicar los nodos `Apriori(3.7)` o `FPGrowth(3.7)`. Observarás que las mejores reglas encontradas son entre productos que no se han comprado (valor 0).
- Parte (c): Para encontrar reglas entre productos que sí se han comprado (valor 1), debes transformar la base de datos relacional en una transaccional. Es decir, debes 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 tengas construida la base de datos transaccional, puedes aplicar los nodos `Item Set Finder (Borgelt)` y `Association Rule Learner` para encontrar itemsets frecuentes y reglas de asociación importantes, respectivamente.

Puedes construir las transacciones como deseos, 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`.

- Parte (d): Identifica itemsets frecuentes y reglas de asociación importantes. Ajusta el soporte y la confianza mínimos para obtener reglas que contengan al menos 2 items en el antecedente.
- Parte (e): Discute las relaciones interesantes que encuentres.

Solución Obtenida en Knime

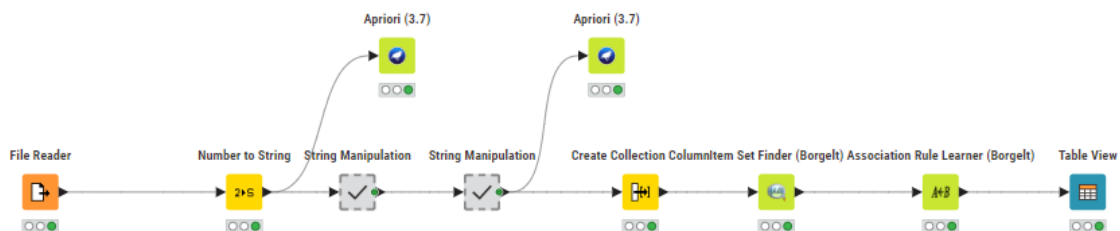


Figura 1: Solución 1

Nodo File Reader

- Cada fila de la tabla representa una transacción con los productos comprados en esa ocasión.
- Un valor de 1 indica que el producto fue comprado, mientras que 0 significa que no lo fue.
- Esta tabla es la base para convertir los datos a un formato transaccional y realizar análisis de asociaciones más adelante en el flujo.

El siguiente nodo en el flujo de trabajo es el Number to String, que se utilizará para transformar los valores numéricos en cadenas de texto, facilitando su manipulación en los nodos posteriores.

Nodo Number to String

Se utiliza el formato predeterminado para la conversión de números a cadenas. Los valores 1 y 0 permanecen como texto, pero ahora son tratables como strings en lugar de enteros. La salida es una tabla donde los valores de las columnas originalmente numéricas se transforman en cadenas. Aunque visualmente no hay cambios evidentes, los datos ahora son del tipo String.

Nodo Apriori 3.7

Configuración del Nodo:

- car: False. No se busca generar reglas asociativas de tipo class association rules.
- classIndex: -1. No hay atributo objetivo (todas las columnas se consideran por igual).
- delta: 0.05. Incremento en el soporte mínimo durante las iteraciones del algoritmo.
- lowerBoundMinSupport: 0.05. Soporte mínimo inicial establecido en 5 %. Los conjuntos de ítems con soporte menor a este serán descartados.
- metricType: Confidence. Las reglas generadas se evalúan según su confianza.
- minMetric: 0.8. Se filtran las reglas con una confianza menor a 80 %.
- numRules: 10. El número máximo de reglas a mostrar es 10.
- outputItemSets: False. No se muestran los conjuntos de ítems generados, solo las reglas finales.
- removeAllMissingCols: False. Las columnas con valores perdidos no se eliminan.
- significanceLevel: -1.0. No se utiliza ninguna prueba estadística para evaluar la significancia de las reglas.
- treatZeroAsMissing: False. Los valores 0 no se interpretan como valores perdidos.
- upperBoundMinSupport: 1.0. El soporte máximo es del 100
- verbose: False. No se generan mensajes detallados en la consola.

Salida del Nodo:

- Soporte mínimo: 0.2 (93 instancias). Las reglas y conjuntos de ítems deben aparecer en al menos el 20 % de las instancias.

- Número de ciclos realizados: 16. Esto indica las iteraciones necesarias para encontrar las reglas.
- Conjuntos de ítems grandes generados:
 - L(1) (Ítems individuales): 44.
 - L(2) (Combinaciones de 2 ítems): 638.
 - L(3) (Combinaciones de 3 ítems): 1540.
 - L(4) (Combinaciones de 4 ítems): 1509.

Mejores reglas encontradas:

1. Regla: Hazelnut=0, Carrot=0, Onion=0 \rightarrow Toothpaste=0]
 - Confianza: 0.88.
 - Levantamiento (Lift): 1.43.
 - Interpretación: Si Hazelnut, Carrot y Onion no están presentes, entonces Toothpaste probablemente tampoco lo estará (88% de confianza).
2. Regla: Hazelnut=0, Onion=0, Flour=0 \rightarrow Toothpaste=0
 - Confianza: 0.88.
 - Levantamiento (Lift): 1.43.
3. Regla: Apple=0, Hazelnut=0, Onion=0 \rightarrow Toothpaste=0
 - Confianza: 0.86.
 - Levantamiento (Lift): 1.39.
4. Regla: ShavingFoam=0, Salt=0, Flour=0 \rightarrow Toothpaste=0
 - Confianza: 0.86.
 - Levantamiento (Lift): 1.39.
5. Regla: Bread=0, Apple=0, Cheese=0 \rightarrow Butter=0
 - Confianza: 0.85.
 - Levantamiento (Lift): 1.36.

Nodo Metanodo 1

Este meta-nodo transforma todos los valores 0 de las columnas de productos en valores perdidos (null). Esto permite trabajar únicamente con los productos comprados (1) en los siguientes pasos del flujo de trabajo.



Figura 2: Metanodo 1

Nodos del Metanodo:

Nodo String Manipulation:

Hay un String Manipulation por cada columna y cada nodo se encarga de poner a *null* los valores de la columna que se encuentran a '0', para esto se utiliza la siguiente expresión:

```
1 replace($Nombre_columna$, "0", "null")
2 Replace Column: Nombre_columna
```

Nodo String to Number:

Convierte las columnas modificadas de tipo String a tipo Number para que las herramientas puedan interpretar los valores *null* como valores perdidos (?).

Nodo Number to String:

Convierte nuevamente las columnas de tipo Number a tipo String. Esto asegura que las salidas sean compatibles con pasos posteriores como la creación de transacciones.

Entrada de los datos:

Rows: 464 | Columns: 22

Table Statistics

Bread String	Honey String	Bacon String	Toothpaste String	Banana String	Apple String	Hazelnut String	Cheese String	Meat String	Carrot String	Cucumber String	Onion String	Milk String
1	0	1	0	1	1	1	0	0	1	0	0	0
1	1	1	0	1	1	1	0	0	0	1	0	1
0	1	1	1	1	1	1	1	1	0	1	1	1
1	1	0	1	0	1	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	0	1	0	0	0	0	1	0
0	0	1	0	1	1	0	0	0	1	0	0	0
0	0	1	1	1	0	1	0	0	0	1	1	0
0	1	1	0	1	1	1	1	1	1	0	1	0
0	0	0	0	0	0	0	0	0	0	0	1	1
0	0	1	1	0	1	1	0	0	1	0	0	0

Figura 3: Etrada Metanodo 1

Salida de los datos:

Rows: 464 | Columns: 22

Table Statistics

Bread String	Honey String	Bacon String	Toothpaste String	Banana String	Apple String	Hazelnut String	Cheese String	Meat String	Carrot String	Cucumber String	Onion String	Milk String
1	⓪	1	⓪	1	1	1	⓪	⓪	1	⓪	⓪	⓪
1	1	1	⓪	1	1	1	⓪	⓪	⓪	1	⓪	1
⓪	1	1	1	1	1	1	1	1	⓪	1	1	1
1	1	⓪	1	⓪	1	⓪	⓪	⓪	⓪	1	1	1
⓪	1	⓪	⓪	⓪	⓪	⓪	⓪	⓪	⓪	⓪	⓪	⓪
⓪	1	⓪	1	⓪	⓪	1	⓪	⓪	⓪	⓪	1	⓪
⓪	⓪	1	⓪	1	1	⓪	⓪	⓪	1	⓪	⓪	⓪
⓪	⓪	1	1	1	⓪	1	⓪	⓪	⓪	1	1	⓪
⓪	1	1	⓪	1	1	1	1	1	1	⓪	1	⓪
⓪	⓪	⓪	⓪	⓪	⓪	⓪	⓪	⓪	⓪	⓪	1	1
⓪	⓪	1	1	⓪	1	1	⓪	⓪	1	⓪	⓪	⓪

Figura 4: Salida Metanodo 1

Nodo Metanodo 2

Este meta-nodo reemplaza los valores 1 de cada columna de la tabla por el nombre de la columna correspondiente. Esto transforma las compras en una representación textual que facilita la creación de transacciones.

Nodos del Metanodo:

Nodo String Manipulation:

Hay un String Manipulation por cada columna y cada uno reemplaza los '1' por el nombre de la columna en cada una de las columnas de productos. Para esto se utiliza la siguiente expresión:

```
1 replace($Nombre_Columna$, "1", "Nombre_columna")
2 Replace Column: Nombre_Columna
```

[H] Salida del Metanodo 2:

Nodo Apriori 3.7

Tras convertir los valores 1 en nombres de productos y aplicar nuevamente el nodo Apriori, se observan diferencias significativas en los resultados debido a que ahora se consideran exclusivamente los

Rows: 464 | Columns: 22

Table Statistics

Bread String	Honey String	Bacon String	Toothpaste String	Banana String	Apple String	Hazelnut String	Cheese String	Meat String	Carrot String	Cucumber String	Onion String	Milk String
Bread	?	Bacon	?	Banana	Apple	Hazelnut	?	?	Carrot	?	?	?
Bread	Honey	Bacon	?	Banana	Apple	Hazelnut	?	?	?	Cucumber	?	Milk
?	Honey	Bacon	Toothpaste	Banana	Apple	Hazelnut	Cheese	Meat	?	Cucumber	Onion	Milk
Bread	Honey	?	Toothpaste	?	Apple	?	?	?	?	Cucumber	Onion	Milk
?	Honey	?	?	?	?	?	?	?	?	?	?	?
?	Honey	?	Toothpaste	?	?	Hazelnut	?	?	?	?	Onion	?
?	?	Bacon	?	Banana	Apple	?	?	?	Carrot	?	?	?
?	?	Bacon	Toothpaste	Banana	?	Hazelnut	?	?	?	Cucumber	Onion	?
?	Honey	Bacon	?	Banana	Apple	Hazelnut	Cheese	Meat	Carrot	?	Onion	?
?	?	?	?	?	?	?	?	?	?	?	Onion	Milk
?	?	Bacon	Toothpaste	?	Apple	Hazelnut	?	?	Carrot	?	?	?

Figura 5: Salida Metanodo 2

productos comprados (ignorando las compras no realizadas).

Interpretación de la Salida:

- Soporte mínimo: 0.05 (23 transacciones como mínimo).
- Confianza mínima: 0.8.
- Número máximo de reglas generadas: 10.
- 1-elementos (L(1)): 22. Esto significa que 22 productos individuales alcanzaron el soporte mínimo.
- 2-elementos (L(2)): 231.
- 3-elementos (L(3)): 1540.
- 4-elementos (L(4)): 2320.
- 5-elementos (L(5)): 45.

Las mejores reglas generadas:

1. Regla 1:

- Antecedente: Bacon=Bacon, Banana=Banana, Butter=Butter, Egg=Egg.
- Consecuente: Cheese=Cheese.
- Confianza: 0.93 → 93 % de las transacciones con Bacon, Banana, Butter y Egg también incluyen Cheese.
- Lift: 2.09 → La compra de Cheese es 2.09 veces más probable cuando se compran los productos del antecedente.
- Convicción: 5 → Refleja una fuerte relación causal.

2. Regla 2:

- Antecedente: Honey=Honey, Banana=Banana, Meat=Meat, Onion=Onion.
- Consecuente: Bacon=Bacon.
- Confianza: 0.89 → 89 % de las transacciones con Honey, Banana, Meat y Onion también incluyen Bacon.
- Lift: 2.06 → La compra de Bacon es 2.06 veces más probable con el antecedente.

3. Regla 5:

- Antecedente: Bacon=Bacon, Meat=Meat, Carrot=Carrot, Onion=Onion.
- Consecuente: Honey=Honey.
- Confianza: $0.86 \rightarrow 86\%$ de las transacciones con el antecedente incluyen Honey.
- Lift: $2.07 \rightarrow$ La probabilidad de compra de Honey se duplica con el antecedente.

Cambios Observados:

Conjunto L(1): Ahora hay 22 productos frecuentes en lugar de 44 en la primera ejecución (con productos no comprados). Esto refleja que las columnas con valores 0 no influyen más en el cálculo de soporte. Reglas: Todas las reglas reflejan relaciones entre productos comprados, eliminando combinaciones con productos no comprados (valor 0).

Nodo Create Collection Column

Este nodo transforma la base de datos en un formato más adecuado para el análisis de transacciones, donde cada fila se convierte en una lista de productos comprados (un conjunto de productos en cada transacción). Como resultado, todas las columnas con productos ahora se representan en una sola columna llamada Transaction. Configuración:

- Ignore Missing Value: Los valores nulos o vacíos son ignorados.
- Remove Aggregated Columns: Eliminar las columnas agregadas del conjunto de datos original.
- Nueva columna: Se crea una nueva columna llamada "Transaction", que almacena los productos comprados para cada transacción en formato de lista.

Nodo Item Set Finder (Borgelt)

El nodo Item Set Finder (Borgelt) aplica el algoritmo Apriori para encontrar conjuntos de productos frecuentes en las transacciones, generando reglas de asociación útiles. Este algoritmo busca identificar combinaciones de productos que aparecen con frecuencia en las transacciones de compra, según los parámetros definidos. Configuración

- Algoritmo: Apriori (utilizado para encontrar los conjuntos de elementos frecuentes en los datos).
- Tipo de Objetivo: Frequent (el objetivo es encontrar conjuntos de elementos frecuentes).
- Tamaño Mínimo de Conjunto: 3 (se buscan conjuntos de al menos 3 productos).
- Tamaño Máximo de Conjunto: 10 (se permiten conjuntos con un máximo de 10 productos).
- Soporte Mínimo: 10.0% (esto significa que los conjuntos de productos deben aparecer al menos en el 10% de las transacciones).

Salida:

Rows: 296 | Columns: 4

Table Statistics

#	RowID	ItemSet Set	ItemSetSize Number (integer)	ItemSetSupport Number (integer)	RelativeItemSetSupport% Number (double)
296	Row...	[Bacon,Cheese,Banana]	3	65	14.009
98	Row97	[Butter,Bacon,Cheese]	3	64	13.793
246	Row...	[ShavingFoam,Bacon,Banana]	3	62	13.362
124	Row...	[Onion,Bacon,Banana]	3	61	13.147
261	Row...	[Egg,Bacon,Cheese]	3	61	13.147
263	Row...	[Egg,Cheese,Banana]	3	60	12.931
87	Row86	[Butter,ShavingFoam,Bacon]	3	59	12.716
228	Row...	[Bread,Bacon,Banana]	3	59	12.716
245	Row...	[ShavingFoam,Bacon,Cheese]	3	59	12.716
277	Row...	[Carrot,Cheese,Banana]	3	59	12.716
285	Row...	[Honey,Bacon,Banana]	3	59	12.716
294	Row...	[Hazelnut,Bacon,Banana]	3	59	12.716
147	Row...	[Meat,Honey,Bacon]	3	58	12.5
153	Row...	[Meat,Bacon,Cheese]	3	58	12.5
222	Row...	[Bread,Honey,Cheese]	3	58	12.5
266	Row...	[Carrot,Honey,Cheese]	3	58	12.5

Figura 6: Item Set Finder

Observaciones:

- **Combinaciones frecuentes de productos:** Los conjuntos de productos que aparecen juntos con más frecuencia incluyen productos como Bacon, Cheese, Banana, Butter, Bacon, Cheese, ShavingFoam, Bacon, Banana, y Onion, Bacon, Banana, entre otros. Estas combinaciones indican que los productos mencionados son comúnmente adquiridos juntos por los consumidores.
- **Tendencias comunes:** Muchas de las combinaciones frecuentes incluyen productos populares como Bacon, Cheese, Banana, y ShavingFoam, lo que sugiere que estos productos se compran comúnmente en conjunto. También se observa que productos como Egg, Honey, Carrot, y Olive aparecen en múltiples combinaciones.
- **Frecuencia de aparición:** Cada conjunto tiene un valor de "ItemSetSupport", que muestra el número de transacciones en las que dicho conjunto de productos aparece. Por ejemplo:
 - El conjunto [Bacon, Cheese, Banana] aparece en 65 transacciones.
 - El conjunto [Butter, Bacon, Cheese] aparece en 64 transacciones.

Nodo Association Rule Learner

El nodo se centra en reglas de asociación que describen cómo los productos o elementos de una cesta de compras están relacionados entre sí. Este tipo de análisis ayuda a entender las relaciones de compra entre diferentes productos. Este nodo utiliza el algoritmo Apriori para generar reglas de asociación entre conjuntos de productos. Donde el antecedente es el conjunto de productos que se compran juntos, y el consecuente es el producto que también se compra con mayor probabilidad cuando se compran los productos del antecedente.

Métricas de las reglas de asociación:

- **ItemSetSupport:** Cuanto mayor es este número, mayor es la frecuencia con que el conjunto de productos aparece en el conjunto de datos. Esto es importante porque ayuda a identificar reglas que ocurren con frecuencia en las transacciones.
- **RuleConfidence%:** Indica la fiabilidad de la regla. Una confianza alta significa que cuando se compra el antecedente, la probabilidad de que el consecuente sea comprado también es alta.

- **RuleLift:** Un valor de Lift mayor a 1 indica que la compra del antecedente aumenta la probabilidad de la compra del consecuente. Un Lift bajo o cercano a 1 significa que los productos no están fuertemente relacionados o que la compra del antecedente no influye mucho en la compra del consecuente.
- **Lift > 1:** Los productos están positivamente correlacionados.
- **Lift = 1:** No hay relación entre los productos.
- **Lift < 1:** Los productos están negativamente correlacionados.

Observaciones de las reglas generadas:

Regla 1

- Consequent: Milk
- Antecedent: [HeavyCream]
- ItemSetSupport: 8
- RelativeItemSetSupport %: 2.7027
- RuleConfidence %: 17.8
- RuleLift: 2.7696
- AbsoluteBodySetSupport: 276.96
- RelativeBodySetSupport %: 19.0
- AbsoluteHeadItemSupport: 6.4189

Interpretación:

- La regla dice que si un cliente compra "HeavyCream", hay una alta probabilidad (17.8% de confianza) de que también compre Milk.
- El Lift de 2.7696 sugiere que la compra de "HeavyCream" aumenta significativamente la probabilidad de comprar "Milk". Esta es una regla fuerte y valiosa para promociones cruzadas.

Regla 2

- Consequent: Sugar
- Antecedent: [Salt]
- ItemSetSupport: 4
- RelativeItemSetSupport %: 1.3514
- RuleConfidence %: 12.5
- RuleLift: 2.6429
- AbsoluteBodySetSupport: 264.29
- RelativeBodySetSupport %: 14.0

- AbsoluteHeadItemSupport: 4.7297

Interpretación:

- Si un cliente compra Salt, hay un 12.5 % de confianza de que también comprará Sugar.
- El Lift de 2.6429 indica una relación positiva entre Salt y Sugar, lo que significa que la compra de Salt aumenta la probabilidad de comprar Sugar de manera significativa.

Regla 3

- Consequent: Cucumber
- Antecedent: [Banana]
- ItemSetSupport: 10
- RelativeItemSetSupport %: 3.3784
- RuleConfidence %: 10.1
- RuleLift: 1.8687
- AbsoluteBodySetSupport: 99.0
- RelativeBodySetSupport %: 33.4
- AbsoluteHeadItemSupport: 16.0

Interpretación:

- Si un cliente compra Banana, hay un 10.1 % de probabilidad de que también compre Cucumber.
- El Lift de 1.8687 indica una correlación positiva moderada entre Banana y Cucumber, lo que hace que esta regla sea útil para agrupar estos productos en promociones o recomendaciones.

Regla 4

- Consequent: Apple
- Antecedent: [Banana]
- ItemSetSupport: 15
- RelativeItemSetSupport %: 5.0676
- RuleConfidence %: 15.2
- RuleLift: 1.6611
- AbsoluteBodySetSupport: 99.0
- RelativeBodySetSupport %: 33.4
- AbsoluteHeadItemSupport: 27.0

Interpretación:

- Si un cliente compra Banana, hay un 15.2% de probabilidad de que también compre Apple.
- El Lift de 1.6611 muestra una correlación positiva entre Banana y Apple, indicando que los consumidores que compran Banana a menudo también compran Apple.

Las reglas generadas por el nodo Association Rule Learner muestran relaciones entre productos. Las columnas clave a observar para evaluar la importancia de las reglas son Lift y Confidence. Las reglas con un Lift mayor a 1 indican que los productos tienen una relación positiva, y las de alta confianza muestran que los productos en el antecedente se compran con frecuencia junto al consecuente.

De todas las reglas, la más importante es la Regla 1 con Milk y HeavyCream, ya que tiene un Lift alto (2.7696) y es confiable, lo que sugiere una fuerte relación entre ambos productos.

Ejercicio 2

Enunciado

Cesta de la Compra (4 puntos)

Considera el dataset contenido en el fichero `datos_compras.csv`, el cual representa la cesta de la compra de clientes de un supermercado. Cada fila corresponde a una transacción, indicando los productos comprados por un cliente. Realiza un análisis exhaustivo para identificar patrones y relaciones en los datos siguiendo las instrucciones a continuación:

1. Lectura y preprocesamiento de los datos:

- Lee los datos utilizando el nodo *File Reader (Complex Format)*. Este nodo generará una columna tipo `string` con los productos delimitados por comas.
- Utiliza el nodo *Cell Splitter* para dividir los productos en columnas separadas, tomando como delimitador la coma.
- Crea una nueva columna que agrupe los productos comprados en un conjunto transaccional mediante el nodo *Create Collection Column*, ignorando los valores perdidos.

2. Identificación de itemsets frecuentes:

- Utiliza el nodo *Item Set Finder (Borgelt)* para encontrar conjuntos de productos frecuentes en las transacciones. Ajusta los parámetros del nodo:
 - Algoritmo: *Apriori*.
 - Soporte mínimo: 5%.
 - Tamaño mínimo y máximo de los conjuntos: entre 1 y 10.

3. Generación de reglas de asociación:

- Aplica el nodo *Association Rule Learner* para generar reglas de asociación que describan las relaciones entre productos. Ajusta los umbrales de soporte, confianza y lift para obtener reglas significativas.

4. Exploración de reglas negativas:

- Analiza reglas del tipo:
 - *No item1* \rightarrow *No item2*: La ausencia de un producto predice la ausencia de otro.
 - *item1* \rightarrow *No item2*: La compra de un producto reduce la probabilidad de comprar otro.

- $No\ item1 \rightarrow item2$: La ausencia de un producto incrementa la probabilidad de comprar otro.

5. Estrategias de marketing:

- Basándote en los resultados obtenidos, propone estrategias de marketing que el supermercado podría implementar para incrementar las ventas. Considera recomendaciones de productos, promociones cruzadas y agrupación de productos en la tienda.

Solución KNIME

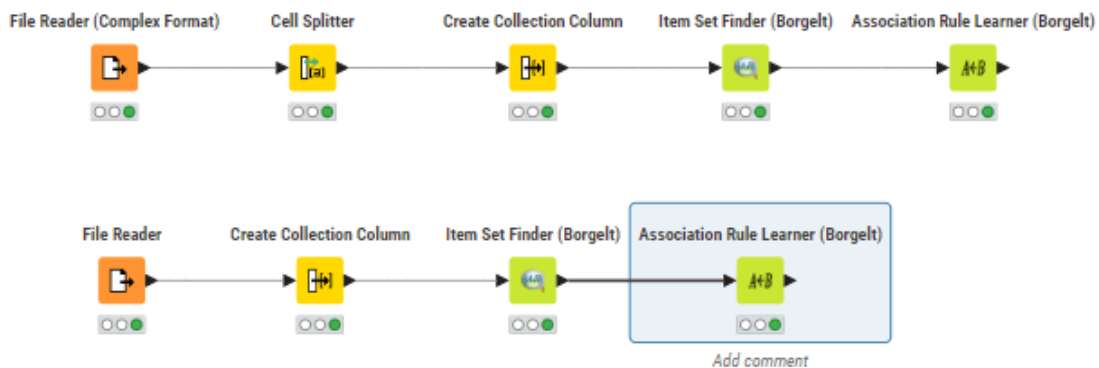


Figura 7: Solución

Nodo File Reader (Complex Format)

Permite leer el archivo CSV con las transacciones de compra. Se utiliza el formato complejo para manejar columnas con datos no estructurados o delimitados. El nodo genera una columna única de tipo *string*, donde cada valor contiene una lista de productos separados por comas.

Configuración:

- Seleccionar el archivo `datos_compras.csv`.
- Configurar como delimitador la coma (,).
- Asegurarse de que las filas se carguen correctamente como cadenas de texto.

Salida: Columna única donde cada fila contiene los productos comprados por un cliente, separados por comas. Ejemplo:

Rows: 7501 | Columns: 1

Table Statistics

#	RowID	Col0 String
1	Row0	shrimp,almonds,avocado,vegetables mix,green grapes,whole weat flour,yams,cottage cheese,energy drink,tomato juice,low fat yogurt,green tea,honey,salad,mineral water,salmon,antioxydant ju...
2	Row1	burgers,meatballs,eggs
3	Row2	chutney
4	Row3	turkey,avocado
5	Row4	mineral water,milk,energy bar,whole wheat rice,green tea
6	Row5	low fat yogurt
7	Row6	whole wheat pasta,french fries
8	Row7	soup,light cream,shallot
9	Row8	frozen vegetables,spaghetti,green tea
10	Row9	french fries
11	Row10	eggs,pet food
12	Row11	cookies
13	Row12	turkey,burgers,mineral water,eggs,cooking oil
14	Row13	spaghetti,champagne,cookies

Figura 8: File Reader

Nodo Cell Spliter

Este nodo divide los valores de una columna en varias columnas, utilizando un delimitador. En este caso, se separaron los productos por comas, creando columnas independientes para cada producto.

Configuración:

- Seleccionar la columna que contiene las cadenas de productos.
- Configurar el delimitador como , (coma).
- Indicar que se generen múltiples columnas.

Salida: Cada producto se almacena en una columna separada. Si un cliente compró tres productos, la fila tendrá tres columnas con los nombres de los productos. Ejemplo:

Rows: 7501 | Columns: 21

Table Statistics

#	RowID	Col0 String	Col0_Arr[0] String	Col0_Arr[1] String	Col0_Arr[2] String	Col0_Arr[3] String	Col0_Arr[4] String	Col0_Arr[5] String	Col0_Arr[6] String	Col0_Arr[7] String	Col0_Arr[8] String	Col0_Arr[9] String	C St
1	Row0	shrimp,almon...	shrimp	almonds	avocado	vegetables mix	green grapes	whole weat fl...	yams	cottage cheese	energy drink	tomato juice	low fa
2	Row1	burgers,meat...	burgers	meatballs	eggs								
3	Row2	chutney	chutney										
4	Row3	turkey,avocado	turkey	avocado									
5	Row4	mineral water,...	mineral water	milk	energy bar	whole wheat r...	green tea						
6	Row5	low fat yogurt											
7	Row6	whole wheat ...	whole wheat ...	french fries									
8	Row7	soup,light cre...	soup	light cream	shallot								
9	Row8	frozen vegeta...	frozen vegeta...	spaghetti	green tea								
10	Row9	french fries	french fries										
11	Row10	eggs,pet food	eggs	pet food									
12	Row11	cookies	cookies										
13	Row12	turkey,burgers...	turkey	burgers	mineral water	eggs	cooking oil						

Figura 9: Cell Splitter

Nodo Create Collection Column

Este nodo crea una nueva columna donde se agrupan los valores de múltiples columnas en una lista, ignorando los valores perdidos. Esto transforma los datos en formato transaccional.

Configuración:

- Seleccionar todas las columnas que contienen productos como entrada.
- Activar la opción de “Ignorar valores perdidos” para evitar incluir productos no comprados en las transacciones.
- Asignar un nombre a la nueva columna, por ejemplo, **Transaction**.

Salida: Una columna con listas de productos comprados en cada transacción. Ejemplo:

<input type="checkbox"/>	#	RowID	Transaction Set
<input type="checkbox"/>	1	Row0	[shrimp,almonds,avocado,vegetables mix,green grapes,whole wheat flour,yams,cottage cheese,energy drink,tomato juice,low fat yogurt,green tea,honey,salad,mineral water,salmon,antioxydant]...
<input type="checkbox"/>	2	Row1	[burgers,meatballs,eggs,burgers,meatballs,...]
<input type="checkbox"/>	3	Row2	[chutney]
<input type="checkbox"/>	4	Row3	[turkey,avocado,turkey,avocado]
<input type="checkbox"/>	5	Row4	[mineral water,milk,energy bar,whole wheat rice,green tea,mineral water,milk,...]
<input type="checkbox"/>	6	Row5	[low fat yogurt]
<input type="checkbox"/>	7	Row6	[whole wheat pasta,french fries,whole wheat pasta,french fries]
<input type="checkbox"/>	8	Row7	[soup,light cream,shallot,soup,light cream,...]
<input type="checkbox"/>	9	Row8	[frozen vegetables,spaghetti,green tea,frozen vegetables,spaghetti,...]
<input type="checkbox"/>	10	Row9	[french fries]
<input type="checkbox"/>	11	Row10	[eggs,pet food,eggs,pet food]
<input type="checkbox"/>	12	Row11	[cookies]
<input type="checkbox"/>	13	Row12	[turkey,burgers,mineral water,eggs,cooking oil,turkey,burgers,...]
<input type="checkbox"/>	14	Row13	[spaghetti,champagne,cookies,spaghetti,champagne,...]

Figura 10: Create Collection Column

Nodo Item Set Finder (Borgelt)

Este nodo utiliza el algoritmo *Apriori* para encontrar conjuntos de productos que aparecen juntos con cierta frecuencia en las transacciones.

Configuración:

- **Algoritmo:** *Apriori*.
- **Soporte mínimo:** 5 % (para identificar conjuntos relevantes que ocurren al menos en un 5 % de las transacciones).
- **Tamaño mínimo y máximo del conjunto:** Configurado entre 1 y 10.

Salida:

- **ItemSet:** Conjunto de productos que aparecen juntos.
- **ItemSetSupport:** Frecuencia absoluta de ocurrencia del conjunto.
- **RelativeItemSetSupport %:** Frecuencia relativa del conjunto, expresada como porcentaje.

Ejemplo:

Rows: 28 | Columns: 4

Table Statistics

#	RowID	ItemSet Set	ItemSetSize Number (integer)	ItemSetSupport Number (integer)	RelativeItemSetSupport% Number (double) ↓
<input type="checkbox"/>	28	Row27 [mineral water]	1	1788	23.837
<input type="checkbox"/>	24	Row23 [eggs]	1	1348	17.971
<input type="checkbox"/>	26	Row25 [spaghetti]	1	1306	17.411
<input type="checkbox"/>	21	Row20 [french fries]	1	1282	17.091
<input type="checkbox"/>	22	Row21 [chocolate]	1	1229	16.384
<input type="checkbox"/>	19	Row18 [green tea]	1	991	13.212
<input type="checkbox"/>	20	Row19 [milk]	1	972	12.958
<input type="checkbox"/>	18	Row17 [ground beef]	1	737	9.825
<input type="checkbox"/>	17	Row16 [frozen vegetables]	1	715	9.532
<input type="checkbox"/>	16	Row15 [pancakes]	1	713	9.505
<input type="checkbox"/>	15	Row14 [burgers]	1	654	8.719
<input type="checkbox"/>	14	Row13 [cake]	1	608	8.106
<input type="checkbox"/>	1	Row0 [cookies]	1	603	8.039
<input type="checkbox"/>	6	Row5 [escalope]	1	595	7.932
<input type="checkbox"/>	10	Row9 [low fat yogurt]	1	574	7.652
<input type="checkbox"/>	13	Row12 [shrimp]	1	536	7.146

Figura 11: Item Set Finder

Nodo Association Rule Learner (Borgelt)

Descripción: Este nodo genera reglas de asociación a partir de los conjuntos frecuentes identificados en el nodo anterior. Evalúa la relación entre productos mediante métricas como confianza (*confidence*) y elevación (*lift*).

Configuración:

- Usar como entrada los conjuntos frecuentes del **Item Set Finder**.
- Ajustar los umbrales de confianza y soporte para filtrar reglas.
- Configurar la métrica de ordenación de reglas como *RuleLift*.

Salida:

- **Antecedent:** Productos que implican la compra de otro producto.
- **Consequent:** Producto cuya compra es implicada.
- **RuleConfidence%:** Porcentaje de fiabilidad de la regla.
- **RuleLift:** Indicador de la fortaleza de la asociación entre antecedente y consecuente.

Ejemplo:

#	RowID	Consequent String	Antecedent Set	ItemSetSu... Number (integ...	Relativit... Number (doub...	RuleConfi... Number (doub...	AbsoluteB... Number (doub...	RelativeB... Number (doub...	RuleLift Number (doub...	RuleLift% Number (doub...	AbsoluteH... Number (doub...	RelativeH... Number (doub...
<input type="checkbox"/>	1	Row0 chocolate	[mineral water]	1	3.571	25	4	14.3	3.5	350	2	7.143
<input type="checkbox"/>	2	Row1 eggs	[mineral water]	1	3.571	25	4	14.3	3.5	350	2	7.143
<input type="checkbox"/>	3	Row2 spaghetti	[mineral water]	1	3.571	25	4	14.3	3.5	350	2	7.143
<input type="checkbox"/>	4	Row3 mineral water	⊕	4	14.286	14.3	28	100	1	100	4	14.286

Figura 12: Association Rule Learner

Identificación de Itemsets Frecuentes

Se empleó el nodo **Item Set Finder** (Borgelt) con los siguientes parámetros:

- Algoritmo: **Apriori**.
- Soporte mínimo: 5 %.
- Tamaño mínimo y máximo de conjuntos: de 1 a 10 productos.

Resultado: El nodo identificó los conjuntos de productos que se compran juntos con mayor frecuencia. Por ejemplo:

- [eggs] con soporte absoluto de 1348 (17.97 %).
- [mineral water, spaghetti] con soporte absoluto de 448 (5.97 %).

Generación de Reglas de Asociación

Se utilizaron los conjuntos frecuentes generados en el nodo **Item Set Finder** como entrada para el nodo **Association Rule Learner**. Este nodo generó reglas de asociación significativas basadas en soporte, confianza y elevación.

Ejemplo de reglas:

- [mineral water] \rightarrow [chocolate]:
 - **Confianza:** 25.0 %
 - **Lift:** 3.5 (relación positiva fuerte).
- [mineral water] \rightarrow [eggs]:
 - **Confianza:** 25.0 %
 - **Lift:** 3.5 (relación positiva fuerte).

Exploración de Reglas Negativas

Se analizaron reglas negativas mediante la configuración del nodo **Association Rule Learner** para incluir:

- *No item1* \rightarrow *No item2*: Ausencia de productos relacionados.
- *item1* \rightarrow *No item2*: Compra de un producto reduce la probabilidad de otro.
- *No item1* \rightarrow *item2*: Ausencia de un producto aumenta la probabilidad de otro.

Resultado: Por ejemplo:

- **No mineral water** \rightarrow **No chocolate**: Relación negativa que indica que la ausencia de **mineral water** reduce la probabilidad de comprar **chocolate**.
- **eggs** \rightarrow **No spaghetti**: Los clientes que compran **eggs** tienden a no comprar **spaghetti**.

Propuestas de Estrategias de Marketing

Con base en los resultados, se sugieren las siguientes estrategias:

- **Promociones cruzadas:** Ofrecer descuentos en **chocolate** para clientes que compren **mineral water**, ya que ambas están positivamente correlacionadas (**Lift** = 3.5).
- **Ubicación en tienda:** Agrupar productos como **spaghetti** y **mineral water** en la misma sección, dado su alto soporte conjunto.
- **Segmentación de clientes:** Identificar clientes que compran **mineral water** pero no **chocolate**, y enviar promociones para incentivar la compra del segundo producto.