




# De Datos a Decisiones: Viabilidad del Aprendizaje de Asociación de Consumos Médicos en el Ámbito de las Obras Sociales \*

Adriano M. Lusso<sup>1</sup> , Lara V. Acuña Bravo<sup>1</sup> , and Sandra E. Roger<sup>1</sup> 

<sup>1</sup> GILIA, Facultad de Informática. Universidad Nacional del Comahue  
adriano.lusso@est.fi.uncoma.edu.ar roger@fi.uncoma.edu.ar

<sup>2</sup> GHISCO, Facultad de Informática. Universidad Nacional del Comahue  
lara.acuna@est.fi.uncoma.edu.ar

**Resumen** A partir de una amplia cantidad de datos provenientes del Servicio de Obra Social de la Universidad Nacional del Comahue (SOSUNC), en los cuales se describen consumos de sus afiliados respecto de práctica médicas y medicamentos, se realiza este trabajo con el propósito de analizar asociaciones entre dichos consumos. La principal motivación es descubrir relaciones entre diversas patologías médicas, que puede ayudar a agrupar ciertos consumos en planes de afiliados de forma más coherente y compacta, asistiendo así a la toma de decisiones de SOSUNC. Para analizar las asociaciones se utiliza el lenguaje de programación R<sup>3</sup>, junto con el paquete *arules*<sup>4</sup> que proporciona métodos para realizar la minería. Además, se utilizan otros paquetes para una exploración gráfica y de rendimiento de los algoritmos para llevar a cabo los análisis propuestos y profundizar en la utilidad de la elección de estas herramientas.

**Keywords:** minería, reglas de asociación, itemsets frecuentes, rendimiento, gráficos, análisis exploratorio

## 1. Introducción y Motivación

La minería de datos proporciona herramientas de gran utilidad a la hora de encontrar patrones subyacentes en grandes conjuntos de datos [1]. Estas logran aprender las correlaciones intrínsecas entre unidades de datos, sirviendo para sus usuarios como una capa de abstracción entre las complejas técnicas matemáticas y estadísticas que se aplican.

Debido a esto es que las herramientas de minería de datos son actualmente estudiadas para el ámbito de la salud [2]. Tanto hospitales, obras sociales y

---

\* Financiado por la Universidad Nacional del Comahue en el contexto del Proyecto de Investigación *04-F020 Tecnologías Semánticas para el Desarrollo de Agentes Inteligentes* y del Proyecto de Extensión *Análisis Forense de Pericias Médico-legales asistidas por Sistemas Inteligente: preparación de datos*

<sup>3</sup> <https://www.r-project.org/>

<sup>4</sup> <https://cran.r-project.org/web/packages/arules/index.html>

centros de investigación hacen uso de estas técnicas para encontrar patrones entre enfermedades, condiciones médicas, síntomas y consumos médicos.

De esta manera, la generación de reglas de asociación se presenta como una herramienta del estado del arte que permite encontrar patrones de consumo en transacciones de una base de datos [3]. Esta técnica cuenta con un conjunto de diferentes algoritmos que la implementan, entre los que se destacan *Apriori* y *Eclat* [4,5]. Entre alguno de sus usos en el dominio de interés, se encuentra la generación de asociaciones entre enfermedades para la predicción de las mismas [6], el descubrimiento de conocimiento en una base de datos de pacientes diabéticos [7,8] y como la base algorítmica de un modelo predictivo para datos psicométricos de SCL-90 [9].

En el ámbito local de la ciudad de Neuquén se encuentra el Servicio de Obra Social de la Universidad Nacional del Comahue (SOSUNC). SOSUNC cuenta con una base de datos que almacena información de sus afiliados y sus consumos médicos. El área de Auditoría Médica, encargada de la gestión y evaluación de solicitudes de cobertura médica, decide si una determinada solicitud cumple los requerimientos necesarios para que su costo sea cubierto por SOSUNC. Además, el área trabaja en la creación de diferentes planes de cobertura, que permiten agrupar y generalizar conductas médicas repetidas. Para el análisis de los datos necesario en estas tareas, el equipo de Auditoría Médica aún no hace uso de técnicas avanzadas de minería de datos, pudiendo ser esto un elemento diferenciador en la calidad de las decisiones tomadas.

El objetivo de este trabajo es aplicar diferentes algoritmos de generación de reglas de asociación con la base de datos de SOSUNC y analizar su rendimiento. Para esto, se usará como inspiración las pruebas implementadas por Vlad Robu y Vitor Duarte dos Santos [10], donde se comparan los rendimientos de *Apriori* y *Eclat*. Se realizará un muestreo de generaciones de reglas haciendo uso de diferentes hiperparámetros. Para el análisis de los resultados, se usarán métricas como el tiempo de ejecución promedio, el Soporte, la Confianza y el *Lift*. Las comparaciones finales serán entre los rendimientos obtenidos por *Apriori* y *Eclat* para este dominio, así como con los resultados presentados por Vlad Robu y Vitor Duarte. Esto permitirá concluir respecto a la viabilidad de aplicar estas técnicas en el marco de trabajo del equipo de Auditoría Médica de SOSUNC.

En la Sección 2 se presentan los trabajos relacionados y una introducción a las técnicas utilizadas. En la Sección 3, se introduce los datos de análisis y la preparación de los mismos, junto con el panorama de los experimentos realizados y aclaraciones importantes sobre los algoritmos. Luego, la Sección 4 detalla los resultados y busca concluir sobre los mismos. Finalmente, Sección 5 resume el trabajo y se presentan los trabajos futuros.

## 2. Trabajos relacionados

La minería de datos abarca un gran rango de técnicas que se adecúan a diferentes necesidades y perspectivas. Debido a los rápidos avances en recolección de datos y su naturaleza creciente, resulta evidente la necesidad de extraer informa-

ción y conocimiento a partir de ellos. Un análisis manual ya no es factible, por lo que la minería de datos se aprovecha de algoritmos capaces de procesar grandes volúmenes de datos de forma automática, además de proveer herramientas para el análisis de los mismos. Esto le facilita el trabajo a las organizaciones, permitiéndoles entender mejor sus datos y así poder lograr sus objetivos y mejorar su relación con sus clientes.

Este trabajo se enfoca en el análisis de asociación, cuya utilidad es el descubrimiento de relaciones interesantes escondidas detrás de un gran conjunto de datos. El análisis de reglas de asociación a partir de datos transaccionales fue propuesto en 1993 por Agrawal *et al*[11]. Este enfoque resulta de mucha utilidad a día de hoy. Típicamente se utiliza para encontrar patrones de compras en mercados: aquellos conjuntos de ítems (*itemsets*) que se compran juntos frecuentemente. En el ámbito médico también es ampliamente usado, por ejemplo, para encontrar enfermedades que ocurren juntas en general. La utilidad de determinar *itemsets* frecuentes y reglas de asociación es que con la presencia de un ítem se puede inferir con una gran probabilidad que otros ítems también están presentes. Se presenta como un ejemplo de regla de asociación al siguiente enunciado: “Si un cliente compra vino y pan, usualmente también compra queso”. Pueden existir numerosas reglas, pero interesan aquellas que sean de buena calidad y confiables. Para esto, se utilizan métricas para determinar qué tan interesante es una regla: Soporte, Confianza y Lift.

Si consideramos un conjunto de transacciones, donde cada transacción consta de *itemsets*, el soporte se define como la frecuencia porcentual de aparición de un subconjunto de ítems en el conjunto de transacciones. Con el soporte se restringe la obtención de *itemsets* de forma tal que sólo se consideren aquellos que superen un umbral mínimo. A estos *itemsets* se los denomina frecuentes. Por otro lado, al buscar reglas de asociación interesan aquellas que sean de calidad. La Confianza de una regla  $R = X \rightarrow Y$  se define el porcentaje de casos en los que la regla realmente es correcta. Se calcula como  $conf(R) = \frac{supp(X \cup Y)}{supp(X)}$ , donde *supp* es el Soporte.

La inducción de reglas de asociación implica tratar con una gran cantidad de reglas en bases de datos que contengan muchos ítems diferentes. Existen algoritmos eficientes para reducir el espacio de búsqueda, dos de los más populares son *Apriori* y *Eclat*. *Apriori* emplea una búsqueda por niveles *Bottom Up*, comenzando por *itemsets* de tamaño 1, y aumentándolo progresivamente. Del algoritmo se identifican dos fases: generación de candidatos y poda. Para generar los candidatos, *Apriori* utiliza conocimiento previo de los *k-itemsets* para generar los  $(k+1)$ -*itemsets*. Esta forma de generación funciona gracias a la propiedad *apriori*: Si un *itemset* no satisface el umbral de soporte mínimo, agregar un nuevo ítem a ese conjunto tampoco logra que satisfaga el umbral. En la fase de poda, se examinan los subconjuntos  $k-1$  de los *itemsets* generados y se eliminan aquellos que no satisfagan el umbral. *Eclat*, propuesto por Zaki en 1997[12], emplea una búsqueda *depth-first* a partir de una base de datos vertical, en la que para cada *itemset* se almacenan las transacciones que lo contienen. A partir de las transacciones identificadas para los *k-itemsets* frecuentes, se generan los  $(k+1)$ -*itemsets*

frecuentes mediante la intersección de las transacciones. Una desventaja es que *Eclat* no considera la Confianza como un hiperparámetro a la hora de generar los *itemsets* frecuentes.

Algunos trabajos existentes [10] [13] los han comparado en cuanto a eficiencia, basándose en el tiempo y memoria necesarios para obtener las reglas. Para tener un panorama amplio de comparación se los somete a distintos valores de soporte mínimo, donde un valor más bajo implica más esfuerzo. Robu y dos Santos[10] comparan, además, la cantidad de reglas generadas y la distribución de sus valores de soporte.

En este trabajo exploratorio el enfoque será replicar los experimentos con una base de datos del Servicio de Obra Social de la Universidad Nacional del Comahue (SOSUNC). Con la idea de que SOSUNC pueda aplicar los procedimientos de minería para mejorar su toma de decisiones y aportarle nuevas perspectivas. Además de analizar qué algoritmo tiene mejor rendimiento temporal se explorarán formas de interpretar los datos junto con las ventajas que poseen. También se hará un análisis de las distribuciones de los valores de las medidas de interés: Soporte, Confianza y *Lift*.

Para el análisis de datos y computación estadística uno de los lenguajes más populares es R<sup>5</sup>. Esta herramienta se mantiene como uno de los lenguajes por excelencia en este ámbito. Al ser un software *open-source*, mantiene una comunidad activa y una amplia cantidad de paquetes que permiten reutilizar código. En este trabajo, se utilizará dicho lenguaje, junto con el paquete *arules*<sup>6</sup>, el cual contiene métodos para realizar la minería de reglas de asociación e *itemsets* frecuentes.

El paquete *arules* provee funcionalidad para manipular los *datasets* a modo de transacciones y analizar los *itemsets* y reglas de asociación utilizando los algoritmos *Apriori* y *Eclat*. Para esto último, provee interfaces para realizar la minería de forma rápida, basándose en implementaciones en lenguaje C realizadas por Christian Borgelt<sup>7</sup>, que se ha dedicado a pulir y mejorar la eficiencia de dichos algoritmos.

Para la exploración gráfica, se utilizan los paquetes *arulesViz*<sup>8</sup> y *ggplot2*<sup>9</sup>. El primero provee métodos predefinidos para visualizar reglas de asociación e *itemsets* frecuentes, el segundo provee métodos para crear gráficos de forma libre indicándoles los valores correspondientes a cada eje y permitiendo personalizar distintos aspectos de los mismos.

---

<sup>5</sup> <https://www.r-project.org/>

<sup>6</sup> <https://cran.r-project.org/web/packages/arules/index.html>

<sup>7</sup> <https://borgelt.net/software.html>

<sup>8</sup> <https://cran.r-project.org/web/packages/arulesViz/index.html>

<sup>9</sup> <https://cran.r-project.org/web/packages/ggplot2/index.html>

### 3. Propuesta

#### 3.1. Datos y Pre-procesamiento

Durante el desarrollo de este trabajo se utilizarán los datos de consumos de prácticas médicas. En la base de datos de SOSUNC se mantiene un archivo en formato *csv* que contiene los consumos de los afiliados. Las características de este archivo son las siguientes:

- **id\_afiliado**: *String* alfanumérico para identificar al afiliado. Por ejemplo: *6d0e46f26269df8d87636480fd023c9d*.
- **id\_practica**: *String* con el formato *xx.xx.xx.xx*, donde *x* representa un dígito, para identificar la práctica. Por ejemplo: *12.22.01.01*.
- **nombre\_practica**: *String* que representa el nombre de la práctica. Por ejemplo: COLPOSCOPÍA.
- **fecha**: Fecha en formato *dd/mm/aaaa*. Representa la fecha en la que el afiliado consumió dicha práctica. Por ejemplo: *02/05/2024*.
- **dia**: Entero que representa el día. No está precedido por '0'. Por ejemplo: *2*.
- **mes**: Entero que representa el mes. No está precedido por '0'. Por ejemplo: *5*.
- **año**: Entero que representa el año. Por ejemplo: *2024*

Este *csv* cuenta con 286007 filas. Los consumos más antiguos corresponden al año 2019, y los más recientes a septiembre de 2024. Sin embargo, no se utilizarán todos los registros en su totalidad, se acotarán a un rango de fechas y se limitarán a prácticas de interés relacionadas con diabetes.

Para este experimento se trabaja con los consumos realizados entre las fechas 01/07/2023 y 01/07/2024, lo que resulta en 60489 registros, aproximadamente un 21 % del total. El siguiente paso es obtener sólo los consumos correspondientes a las prácticas de interés relacionadas a la diabetes. SOSUNC cuenta con un *csv* donde se especifican estas 17 practicas. Usando esta información, después de descartar todos los consumos que no pertenezcan a las prácticas de interés el conjunto de registros se reduce a 9525. Con este filtrado se obtienen los registros que serán analizados.

Para poder utilizar los métodos provistos por el paquete *arules*, los datos deben responder a un formato compatible con la clase *transactions*. En la práctica, las transacciones pueden almacenarse en diversos formatos: lista, representación vertical y representación horizontal. El *csv* de los consumos con el que se trabaja presenta un formato similar al vertical, donde cada fila representa el un único consumo hecho por un afiliado en una fecha dada. Este formato no es compatible con el que requiere el paquete *arules*, que trabaja con el formato horizontal (matriz binaria).

A continuación, se describe la transformación de la representación de datos del *csv* al formato horizontal. Una aclaración importante es que se agruparán todos los consumos de cada afiliado, y esto representará una transacción. Es decir, en lugar de tener el concepto de ID de transacción, se tiene el de ID del afiliado. Si se tratara de un supermercado, se agruparían todos los consumos

de cada cliente junto con la fecha en la que se realizaron. Sin embargo, en el contexto de SOSUNC, es inusual que un afiliado consuma más de una practica en un mismo día. Por lo tanto, luego de la transformación cada fila representará un único afiliado, y no habrá dos filas con el mismo.

Entre los 9525 registros existen 2227 afiliados diferentes. Por lo tanto, la matriz también tendrá 2227 filas. Cada práctica de interés de diabetes representará una columna, es decir, la matriz tendrá 17 columnas. Por cada afiliado se observan sus consumos y se completa la matriz de forma tal que una celda  $M_{i,j}$  se interpreta como: “el afiliado  $i$  consumió la práctica  $j$ ” en caso de que al celda tenga el valor *TRUE*. En el Cuadro 1 se visualiza un ejemplo acotado de cómo resulta la transformación. los nombres de las prácticas y los ids de los afiliados fueron recortados. La lectura se realiza de la siguiente manera: “El afiliado 001aa7 consumió las práctica de GLUCEMIA y COLESTEROL HDL”.

**Cuadro 1.** Ejemplo de matriz resultante

	CONSULTA NUTRICIONISTA	GLUCEMIA	COLESTEROL HDL	....
001aa7...	FALSE	TRUE	TRUE	...
003984...	FALSE	TRUE	FALSE	...
004751...	FALSE	TRUE	TRUE	...
...	...	...	...	...

Luego de crear una instancia de la clase *transactions* a partir de la matriz, se ejecuta el método *summary(transacciones)* para visualizar un resumen de los *itemsets* y se observa algo interesante: se reduce a tener 12 columnas en lugar de 17. Lo que ocurre es que automáticamente descarta las columnas que valen en su totalidad *FALSE*, es decir, prácticas jamás consumidas en el rango de fecha otorgado. El detalle de las prácticas se puede observar en el Cuadro 2. Por otro lado, se muestra un resumen de las prácticas más consumidas (ver Cuadro 3), y la distribución de longitudes de prácticas consumidas (ver Cuadro 4.)

### 3.2. Experimentos Realizados

Se hace una comparación de los tiempos de ejecución de *Apriori* y *Eclat* al minar reglas de las transacciones generadas, explicadas en la sección anterior. Se somete a ambos algoritmos a trabajar con distintos valores de soporte: 0,01 %, 0,1 %, 0,5 %, 1 %, 5 %, 10 %, 20 %, 30 %, 50 %. Mientras menor sea el soporte, mayor es la cantidad de *itemsets* con las que deben trabajar, y por ende, mayor esfuerzo emplea su ejecución. Para obtener valores de tiempo estables, se repite 10000 el minado de reglas para cada valor de soporte mínimo, se almacena la sumatoria del tiempo empleado en cada repetición y se promedia para obtener el valor final de cada valor de soporte correspondiente.

Para cada algoritmo y valor de soporte se almacenan los tiempos promediados, las reglas obtenidas y resumen que incluye información de interés acerca de

**Cuadro 2.** Resumen de prácticas de interés de diabetes.

<b>Practica</b>	<b>Estado</b>
CONSULTA NUTRICIONISTA CON ESPECIALIZACIÓN EN DIABETES	<b>RETIRADA</b>
GLUCEMIA (C/U) —(antes)— GLUCEMIA	PRESENTE
COLESTEROL HDL (HDL-C) —(antes)— COLESTEROL HDL	PRESENTE
HEMOGLOBINA GLICOSILADA (Hb A1C) —(antes)— HEMOGLOBINA GLICOSILADA	PRESENTE
MICROALBUMINURIA / ALBUMINA URINARIA —(antes)— MICROALBUMINURIA	PRESENTE
COLESTEROL TOTAL	PRESENTE
CONSULTA ESPECIALISTAS EN DIABETES	PRESENTE
CONSULTA INICIAL CON PLAN NUTRICIONAL	PRESENTE
Consulta oft. a domicilio, más de 3 se adj H.C.	<b>RETIRADA</b>
CONSULTA VESTIDA OFTALMOLOGICA (PEDIATRICA Y DE ADULTO)	PRESENTE
RFG BILATERAL (SIN DESCARTABLE) RETINOFLUORESCENCIA	<b>RETIRADA</b>
RG BILATERAL -RETINOGRAFIA	PRESENTE
TOMOGRAFIA OPTICA DE COHERENCIA POR OJO	<b>RETIRADA</b>
LESIONES MACULARES(EDEMA MACULAR DIABÉTICO)	<b>RETIRADA</b>
RETINOPLASTÍA DIABÉTICA SEVERA (PANFOTOCOAGULACIÓN)2Y1/2	PRESENTE
CREATININA - serica o urinaria	PRESENTE
LDL COLESTEROL (LDL-C)	PRESENTE

**Cuadro 3.** Prácticas más frecuentes.

<b>Práctica</b>	<b>Consumida por cuántos afiliados</b>
GLUCEMIA (C/U)	1511
CREATININA - serica o urinaria	1392
COLESTEROL TOTAL	1263
COLESTEROL HDL (HDL-C)	1229
LDL COLESTEROL (LDL-C)	1170

**Cuadro 4.** Distribución de longitud de cantidad de prácticas consumidas por afiliados.

Longitud de prácticas	Cantidad de transacciones
1	638
2	234
3	131
4	279
5	622
6	240
7	69
8	14

las reglas, como su distribución, medidas de calidad, entre otros. Con todos estos datos se procede a realizar un análisis no sólo comparando su tiempo empleado, sino también las reglas generadas junto a sus medidas de calidad. Los resultados se pueden visualizar en la siguiente sección.

Más allá de analizar eficiencia se busca el lado útil de la minería de reglas de asociación. Un detalle de esto es que se genera una cantidad excesiva de reglas, dejando al analista una ardua tarea de estudiarlas para identificar aquellas que sean interesantes. De aquí surge la utilidad de la visualización, logrando que los datos sean accesibles rápidamente y aumentando su capacidad de comunicar ideas. Con la aplicación de gráficos es más fácil resaltar las tendencias y descubrimientos.

Al haber planteado el rendimiento usando varios valores de soporte, se examinará únicamente las reglas generadas con un soporte mínimo del 0,1.

Se analiza la distribución del Soporte, Confianza y *Lift* para las reglas generadas por cada algoritmo. El *Lift* se define como  $lift(X \rightarrow Y) = \frac{supp(X \cup Y)}{supp(X) \cdot supp(Y)}$ . Un valor mayor a 1 indica una relación fuerte, mientras que un valor igual a 1 indica que la relación es azarosa y la regla no representa un patrón real.

Otro análisis es la comparación de la distribución normalizada de la cantidad de reglas generada por ambos algoritmos, por cada cantidad de ítems en las reglas. La normalización empleada se toma del trabajo de Robu[10]: *min-max*, donde  $\bar{x} = \frac{(x - x_{min})}{x_{max} - x_{min}}$  corresponde a un punto en el rango  $[0, 1]$ . Se normaliza en caso de que la cantidad de reglas generadas por longitud sea diferente en ambos algoritmos, de esta manera puede ser comparable.

Por último, se generan gráficos para visualizar distintas perspectivas sobre las reglas en sí, más que con el propósito de comparar lo generado por los distintos algoritmos. Con estos gráficos se busca exponer la utilidad de las herramientas que facilitan el trabajo de análisis. Nos enfocamos en los siguientes tres gráficos.

**3.2.1. Scatter Plot** Este gráfico se enfoca en las medidas de interés y cómo las reglas pueden tener valores similares. El eje  $x$  representa valores de soporte, el eje  $y$  de confianza. Las reglas se posicionan usando estas coordenadas, lo que permite visualizar la cercanía entre sí. Otro detalle interesante de esta representación es que cada regla tiene un color asociado que representa el *Lift*: mientras más fuerte



el color, mayor el valor. El poder ver rápidamente cómo se distribuyen los valores para tres medidas de interés en un único gráfico resulta muy útil.

**3.2.2. Grouped Matrix** Este gráfico en forma de matriz se enfoca en visualizar reglas según su consecuente o antecedente. Organiza conjuntos grandes de antecedentes en pequeños grupos usando *clustering*. Cada agrupación se ubica en las columnas de la matriz y se puede ver la cantidad de reglas que la componen y los ítems de dichas reglas. Por otro lado, las filas corresponden a ítems del consecuente. Se colocan “globos” en las intersecciones para representar que todas las reglas de la columna tienen como consecuente el ítem de la fila. El tamaño del “globo” aumenta según el valor del promedio de los soportes de las reglas que lo constituyen, mientras que el color aumenta según el valor promedio del *Lift*. Este gráfico es especialmente útil cuando se cuenta con una cantidad elevada de reglas.

**3.2.3. Graph** Este gráfico en forma de grafo se enfoca en mostrar cómo las reglas comparten ítems individuales. Cada regla se enumera y se representa en un nodo, mientras que los ítems son representados por arcos. Los arcos representan la relación con la regla: si es entrante significa que es antecedente, y si es saliente, consecuente. Esta visualización es útil para explorar las reglas de forma más visual. Sin embargo, muchas reglas puede volver el grafo muy cargado, es por esto que se suele limitar a un máximo de 100.

### 3.3. Aclaraciones sobre los algoritmos

Los algoritmos del paquete *arules* tienen unas características que son necesarias aclarar ya que se alejan brevemente de cómo se concibieron originalmente. Para empezar, las reglas minadas tienen la forma  $X \rightarrow Y$ , donde ambos son conjuntos disjuntos de *itemsets* y además  $|Y| \geq 1$ . El paquete, sin embargo, sólo genera reglas con un único ítem en el consecuente ( $|Y| = 1$ ). La justificación para esta decisión es que permitir más ítems en el consecuente lleva a generar muchas más reglas, lo que provoca más costo en la ejecución del algoritmo. Además, el que exista una regla  $x \rightarrow y, z$  implica que existen también reglas más simples:  $x \rightarrow y$ ,  $x \rightarrow z$ . El tener únicamente las dos últimas reglas es suficiente para inferir la misma información, considerándose la primer regla irrelevante.

Cada regla obtenida al aplicar los algoritmos contiene el antecedente de la regla, su consecuente y un vector de medidas de interés, lo que nos permitirá hacer diferentes visualizaciones y comparaciones.

A continuación, se aclaran otros detalles importantes de cada algoritmo, según la implementación.

**3.3.1. Apriori** Para minar algoritmos con *Apriori* se lo invoca de esta manera:

```
apriori(data, parameter, appearance, control, ...)
```

donde:

- **data** refiere a los datos a ser minados, debe ser un objeto de la clase *transactions*.
- **parameter** es una lista de parámetros para controlar los aspectos de la minería. Por ejemplo, el soporte y confianza mínimos deseados, la longitud mínima de las reglas, etc.
- **appearance** es una lista de parámetros para especificar restricciones en las asociaciones. Por ejemplo, excluir algún ítem de aparecer en el lado izquierdo o derecho de la regla.
- **control** es una lista de parámetros para controlar aspectos del algoritmo. Por ejemplo, cómo ordenar los ítems según su frecuencia, minimizar uso de memoria, etc.

En el experimento, se utilizó este llamado:

```
apriori(transacciones , parameter = list(support=
    sup_actual , confidence=0.8, minlen=2, maxtime=0) ,
    control=list(verbose=FALSE))
```

Sin importar qué valor de soporte se use (expresado como *sup\_actual* en el fragmento), para la confianza siempre se utilizará un valor de 0,8 (80%). *minlen=2* se utiliza para que siempre se generen reglas con al menos dos ítems, ya que por defecto el valor es 1, lo que provoca que se generen reglas con un antecedente vacío:  $\{\} \rightarrow \{Y\}$ . *maxtime=0* es un parámetro crucial en este experimento. Cuando se usa un soporte mínimo muy bajo se puede crear un conjunto muy grande de *items* y reglas, provocando un tiempo de ejecución elevado. Para evitar esto, el parámetro *maxtime* tiene por defecto el valor 5, lo que significa que el tiempo empleado para verificar los subconjuntos se limita a, como mucho, 5 segundos. Definir este valor en 0 desactiva este límite, lo cual es determinante para hacer una comparación justa. Por último, el parámetro de control *verbose=FALSE* desactiva los *prints* que reportan el progreso del algoritmo.

Otra aclaración importante es que la implementación incluye mejoras con respecto a la teoría original. Por defecto, organiza las transacciones como un *prefix tree*, lo cual acelera el proceso ya que se agrupan por prefijo y se cuentan más rápidamente. Esta opción se puede desactivar, sin embargo, no fue el caso en este experimento para reflejar un uso “normal” de los métodos. Distinto es, con la opción de *maxtime* que limita la búsqueda sobretudo ante valores muy bajos de soporte mínimo, donde en estos casos si se le quiere exigir al algoritmo.

**3.3.2. Eclat** Para minar algoritmos con *Eclat* se lo invoca de esta manera:

```
eclat(data , parameter , control , ...)
```

donde:

- **data** refiere a los datos a ser minados, debe ser un objeto de la clase *transactions*.

- **parameter** es una lista de parámetros para controlar los aspectos de la minería. Por ejemplo, el soporte mínimos deseado, la longitud mínima de las reglas, etc.
- **control** es una lista de parámetros para controlar aspectos del algoritmo. Por ejemplo, cómo ordenar los ítems según su frecuencia, minimizar uso de memoria, etc.

En el experimento, se utilizó este llamado:

```
eclat(transacciones , supp = sup_actual , control=list(
    verbose=FALSE))
```

A diferencia de *Apriori* no posee parámetros de control por defecto que limitan su tiempo de búsqueda.

Debido a que *Eclat* no genera reglas, sino únicamente *itemsets*, se necesita un método extra: *ruleInduction(itemsets)*. Con la ejecución de esta función se obtienen las reglas, por defecto usa una confianza del 80 %, y el método *ptree*: se crea un *prefix tree* con los soportes de los *itemsets* y a partir de ellos se crean las reglas.

#### 4. Análisis de los Resultados

Los resultados obtenidos son presentados como un conjunto de gráficos y tablas. En la Figura 1, se observan los tiempos de ejecución en micro-segundos para cada soporte mínimo configurado para los algoritmos seleccionados. En ambos se observa una pendiente decreciente a medida que el Soporte mínimo es mayor. Aún así, *Apriori* tiene tiempos de ejecución relativamente menores a los de *Eclat*. Esta última observación no condice con investigaciones previas [10,14,13] en las que *Eclat* lograba mejor rendimiento.

En la Figura 2 se muestran diferentes estadísticas para la generación de reglas con un Soporte mínimo de 0,1 %. Todas las métricas para Soporte, Confianza y *Lift* son iguales entre los algoritmos. Además, como se muestra en la Tabla 5, también comparten la misma cantidad de reglas para los diferentes soportes mínimos. Esto motivó a analizar con mayor detalle que reglas de asociación son generadas. Se observó que, para un mismo Soporte mínimo, ambos algoritmos estaban generando los mismos conjuntos de reglas. Esto se suma como una segunda diferencia respecto a investigaciones previas [10,14], donde los algoritmos generaban diferentes reglas y diferente cantidad de las mismas, siendo *Apriori* el algoritmo que se caracterizaba por generar una mayor cantidad.

Los valores de confianza obtenidos, con una media cercana al 1, indican que las reglas generadas son de alta validez y confianza. Las correlaciones proporcionadas por estas pueden ser tenidas en cuenta como verdades generales en el marco de dominio de SOSUNC. Por otro lado, los valores de *Lift* con una media de 1,75, si bien son superiores a 1, distan de ser tan altos como los presentados en otros trabajos [15]. Esto indica que existe un gran impacto sobre el efectos del antecedente de las reglas en la aparición de su consecuente, pero que esta fuerza esta mucho más limitada a la inversa.

En la Figura 3, se observan las distribuciones normalizadas con *min-max* de la cantidad de elementos por regla. Para ejecuciones con mismo Soporte mínimo y diferente algoritmo, las distribuciones son idénticas, lo que corresponde a lo analizado anteriormente. Entre ejecuciones con diferentes soportes mínimos, las ejecuciones sí cambian. La configuración de Soportes de menor valor acumula una mayor cantidad de reglas con mayor cantidad de elementos. A medida que aumenta el soporte configurado, la distribución se reubica a cantidades menores de elementos por regla.

Se cree que las diferencias respecto a la investigación de referencia en los tiempos de ejecución y conjuntos de reglas generadas son ocasionadas por la cantidad de transacciones y elementos utilizados. La referencia [10], como se menciona en la Sección 2, tiene un total de 52938 transacciones y 26 elementos binarios. En este trabajo, se tienen únicamente 2227 transacciones y 12 elementos binarios. Aquí, al existir menor cantidad de combinaciones posibles para generar los conjuntos de elementos frecuentes, es posible que la búsqueda que hace *Eclat* de estos conjuntos logre ser exhaustiva. De esta manera, *Eclat* podría igualar su conjunto de reglas de asociación generado al de *Apriori*.

En la referencia también se teoriza que el menor tiempo de ejecución logrado por *Eclat* puede estar relacionado a que el mismo genera menos reglas de asociación. Al no realizar una búsqueda exhaustiva de los mismos, el algoritmo podría finalizar antes. En este trabajo, se proporciona evidencia que puede sostener esta afirmación. Al haber generado los mismo conjuntos de reglas, ahora *Apriori* logro un tiempo de ejecución menor a *Eclat*. Esto podría indicar que, efectivamente, la menor generación de reglas de *Eclat* pudo haber condicionado el tiempo de ejecución en la investigación de referencia. Aún así, esto no es evidencia suficiente para afirmar por completo tal suposición, ya que podrían existir otros factores de mayor complejidad que hayan sido ignorados durante este trabajo.

Por otro lado, trabajos como el del Dr. Srinadh [13] y literatura en algoritmos de asociación se afirma que *Eclat* supera la eficiencia de *Apriori* ya que no necesita hacer múltiples lecturas a la base de datos. Sin embargo, la implementación de *Apriori* en el paquete *arules* tiene mejoras internas para hacerlo más eficiente, como fue explicado en la Sección 3.3.1. Esto muy probablemente contribuyó a que tuviera una ejecución más rápida, al menos viéndolo desde el punto de vista de usar el algoritmo *Apriori* sin modificaciones.

En cuanto a la exploración de las reglas, la Figura 4 muestra el gráfico *Scatter Plot* resultante de las reglas con soporte mínimo del 0,1 %. Se puede observar que la tendencia general es creciente en el *Lift* en función a la Confianza. Por otro lado, un gran conjunto de reglas se encuentra en el rango  $[0, 0,2)$  de soporte, y a su vez, no se observan reglas con soporte en el rango  $[0,2, 0,35]$  aproximadamente. *arulesViz* provee una versión interactiva de este gráfico a través de un *widget HTML*, lo que permite más capacidades tales como hacer *zoom* y explorar más libremente los puntos, pero por sobre todo, permite ver qué regla representa cada punto y así poder ver con precisión qué reglas en particular están cerca. En la Figura 5 se puede visualizar un ejemplo.

Otro de los gráficos trabajados se puede apreciar en la Figura 6: el gráfico *Grouped Matrix*. Lo más interesante que se rescata de este gráfico es que rápidamente se puede ver que las únicas prácticas presentes en los consecuentes de las reglas son 5: LDL COLESTEROL (LDL-C), COLESTEROL HDL (HDL-C) —(antes)— COLESTEROL HDL, COLESTEROL TOTAL, CREATININA - serica o urinaria y GLUCEMIA (C/U) —(antes)— GLUCEMIA. Esto puede darle a SOSUNC el pie de enfocarse en estas prácticas. Por otro lado, las reglas con mayor *Lift* se encuentran relacionadas con las dos primeras prácticas mencionadas. Este gráfico también tiene su versión interactiva, útil para hacer *zoom* principalmente, pero no para obtener un detalle mayor. Si bien se puede obtener los valores precisos de soporte y *Lift* de cada grupo, no permite ver el detalle de las reglas e ítems que conforman cada uno.

Para finalizar, se generó un grafo para observar la relación de las reglas con los ítems que las componen. Lamentablemente solo puede manejar un máximo de 100 reglas, por defecto escoge las de mayor *Lift*. Es indispensable acceder a la versión interactiva del mismo ya que con una imagen no basta para obtener mucha información. Podemos destacar que LDL COLESTEROL (LDL-C) es la práctica que en todas las reglas aparece como consecuente (Figura 7), es decir, domina las primeras 100 reglas por su *Lift*. Con la versión interactiva se puede poner foco en determinado ítem, pudiendo ver todas las reglas en las que aparece, y ver una regla en particular con sus detalles (Figura 8).

En el marco de aplicación para SOSUNC, se recomienda la utilización de *Apriori*. Al no tener una cantidad de transacciones lo suficientemente grande, las ventajas que proporciona *Eclat* ante mayores conjuntos de datos no son visibles, por ejemplo, su mejora en tiempos de ejecución. Si bien los tiempos resultantes se miden en micro-segundos, siendo valores apenas perceptibles para las personas, la ejecución múltiple y secuenciada de estos algoritmos podría dejar ver estas diferencias. Esta recomendación podría ser extrapolada a otro tipo de entidades con un tamaño organizacional similar, ya que probablemente la cantidad de datos disponibles que tengan sean similares.

Por otra parte, se recomienda fuertemente utilizar gráficos para enriquecer la exploración de las reglas. En los ejemplos presentados, se descubrieron detalles interesantes rápidamente. Esto aligera mucho la carga al analista, por lo que también se recomienda a SOSUNC y a cualquier otra organización complementar los resultados con gráficos y sus versiones interactivas, que agregan aún más utilidad al estudio.

## 5. Conclusión

En este trabajo, se logró replicar las pruebas hechas en la investigación de referencia de Robu y Vitor Duarte dos Santos[10]. Se obtuvieron resultados similares en algunas métricas, pero contrarios en otras. Esto dio pie a encontrar diferencias respecto al conjunto de datos utilizado en este trabajo, el cual era de mucho menor tamaño. Además, se dio fuerza a la suposición hecha en la investigación referencia, que afirmaba que los tiempos de ejecución de *Eclat* obtenidos

estaban relacionados a la cantidad de reglas generadas. También se evaluó la viabilidad de los algoritmos para ser aplicados en SOSUNC, siendo *Apriori* la recomendación final hecha. Por otro lado, se reforzó la utilidad de complementar el estudio con representaciones gráficas para facilitar el trabajo de análisis y exploración de las reglas generadas mediante algunos gráficos de ejemplo y tendencias que se apreciaron de forma visual sin demasiado esfuerzo.

Para trabajos futuros, se propone seguir investigando sobre las posibles causas de que en este trabajo, a diferencia de la investigación de referencia, *Eclat* tuviera mayor tiempo de ejecución. Suponiendo que la cantidad de transacciones y elementos fueran algunos causantes, sería conveniente evaluar el valor de quiebre para estos parámetros, en el cual *Eclat* pase de ser más a menos eficiente que *Apriori*. También es necesario explorar la ejecución con otros parámetros de configuración de la implementación de *Apriori*, por ejemplo, desactivando el uso del *prefix tree*. Esto permitiría analizar que tanto impacta la mejora propuesta por el autor de las implementaciones, con respecto a cómo fue concebido originalmente el algoritmo. Otra forma de enriquecer la comparación, no realizada en este trabajo, es analizar el uso de memoria, ya que puede verse afectada sobre todo para *datasets* muy grandes y puede dar a lugar a otros factores a tener en cuenta a la hora de elegir un método.

Todos los resultados, gráficos y *widgets* están en el repositorio de este trabajo<sup>10</sup>.

## Referencias

1. N. Jain and V. Srivastava, “Data mining techniques: a survey paper,” *IJRET: International Journal of Research in Engineering and Technology*, vol. 2, no. 11, pp. 2319–1163, 2013.
2. N. Jothi, W. Husain, et al., “Data mining in healthcare—a review,” *Procedia computer science*, vol. 72, pp. 306–313, 2015.
3. T. A. Kumbhare and S. V. Chobe, “An overview of association rule mining algorithms,” *International Journal of Computer Science and Information Technologies*, vol. 5, no. 1, pp. 927–930, 2014.
4. M. D. P. M. A. Kothari, “A survey on eclat based algorithm,”
5. K. S. Kumar and R. M. Chezian, “A survey on association rule mining using apriori algorithm,” *International Journal of Computer Applications*, vol. 45, no. 5, pp. 47–50, 2012.
6. A. B. Rao and J. S. Kiran, “Application of market–basket analysis on healthcare,” *International Journal of System Assurance Engineering and Management*, vol. 14, no. Suppl 4, pp. 924–929, 2023.
7. S. Stilou, P. D. Bamidis, N. Maglaveras, and C. Pappas, “Mining association rules from clinical databases: an intelligent diagnostic process in healthcare,” *Studies in health technology and informatics*, no. 2, pp. 1399–1403, 2001.
8. S. Concaro, L. Sacchi, C. Cerra, P. Fratino, and R. Bellazzi, “Mining healthcare data with temporal association rules: Improvements and assessment for a practical

<sup>10</sup> [https://github.com/AdrianoLusso/AssociationRules\\_for\\_MedicalConsumptions](https://github.com/AdrianoLusso/AssociationRules_for_MedicalConsumptions)

- use,” in *Artificial Intelligence in Medicine: 12th Conference on Artificial Intelligence in Medicine, AIME 2009, Verona, Italy, July 18-22, 2009. Proceedings 12*, pp. 16–25, Springer, 2009.
9. Q. Wang and D. S. Yap, “Association rule algorithm-based prediction model for scl-90 psychological measurement: A comparative analysis before and during the covid-19 pandemic,” *Journal of Computing and Electronic Information Management*, vol. 13, no. 2, pp. 32–36, 2024.
  10. V. Robu and V. D. Dos Santos, “Mining frequent patterns in data using apriori and eclat: A comparison of the algorithm performance and association rule generation,” in *2019 6th International Conference on Systems and Informatics (ICSAI)*, pp. 1478–1481, IEEE, 2019.
  11. T. Agrawal, Rakesh; Imieliński and A. Swami, “Mining association rules between sets of items in large databases,” *ACM SIGMOD Record*, vol. 22, pp. 207–216, 1993.
  12. S. O. M. Zaki, M.J; Parthasarathy and W. Li, “New algorithms for fast discovery of association rules,” *ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, vol. 3, pp. 283–296, 1997.
  13. V. Srinadh, “Evaluation of apriori, fp growth and eclat association rule mining algorithms,” *International Journal of Health Sciences*, vol. 6(S2), pp. 7475–7485, 2022.
  14. C.-H. Chee, J. Jaafar, I. A. Aziz, M. H. Hasan, and W. Yeoh, “Algorithms for frequent itemset mining: a literature review,” *Artificial Intelligence Review*, vol. 52, pp. 2603–2621, 2019.
  15. Y. A. Ünvan, “Market basket analysis with association rules,” *Communications in Statistics-Theory and Methods*, vol. 50, no. 7, pp. 1615–1628, 2021.

## A. Gráficos de los resultados

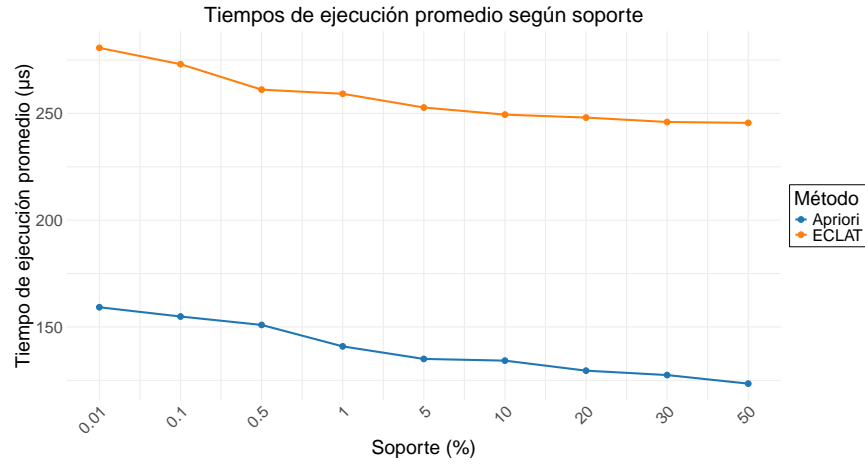


Figura 1. Tiempos de ejecución.

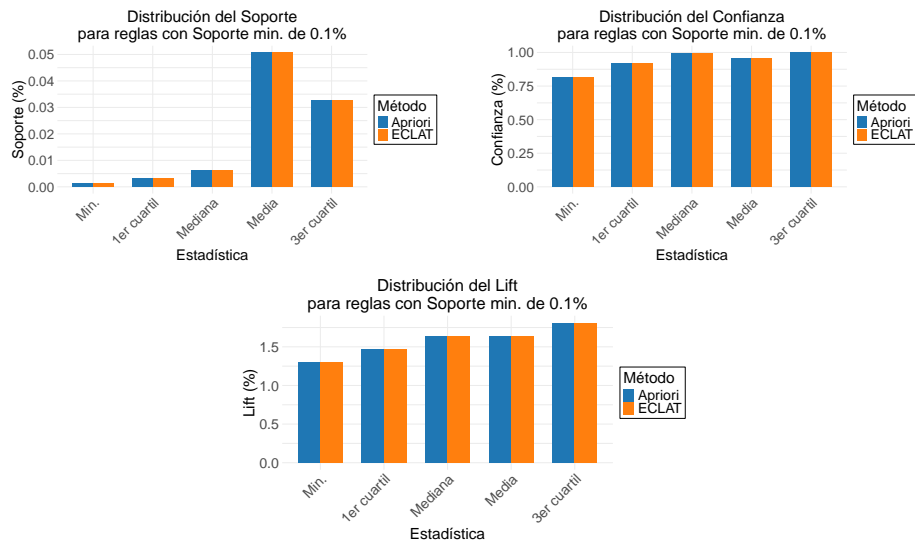
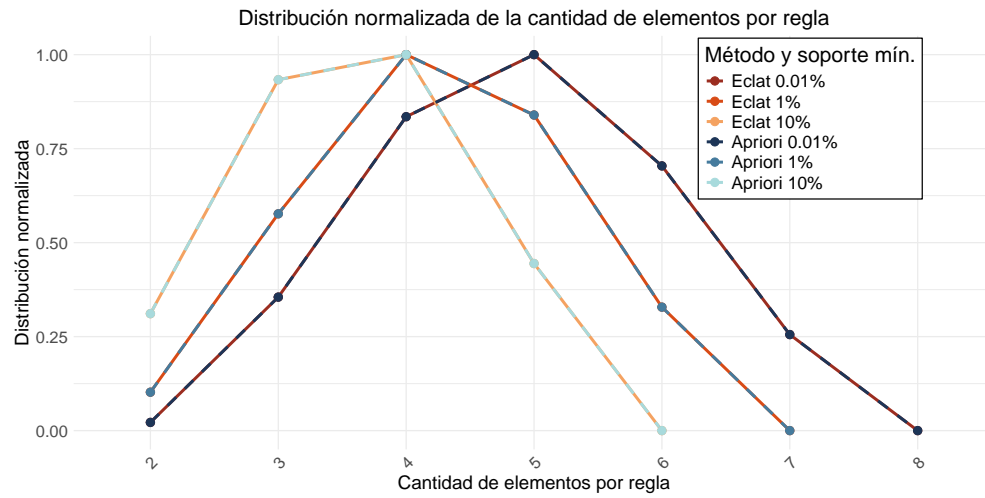


Figura 2. Estadísticas del Soporte, Confianza y *Lift* para la ejecución con Apriori y ECLAT con un Soporte mínimo de 0,1 %.

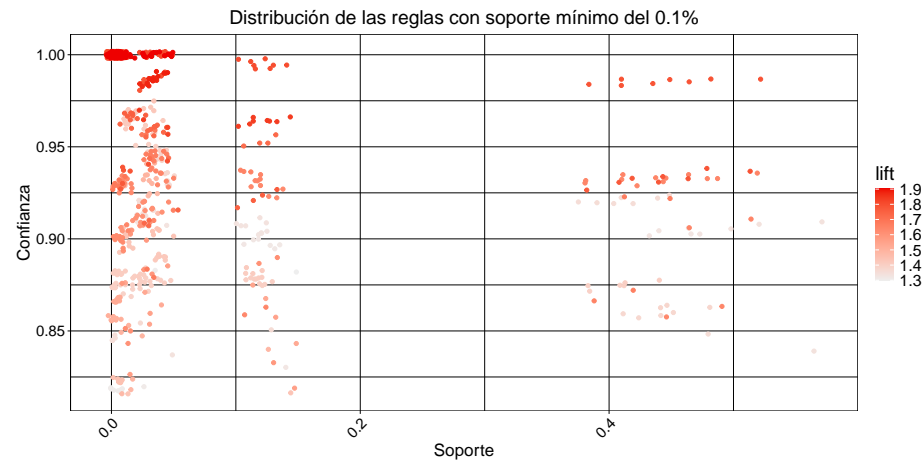




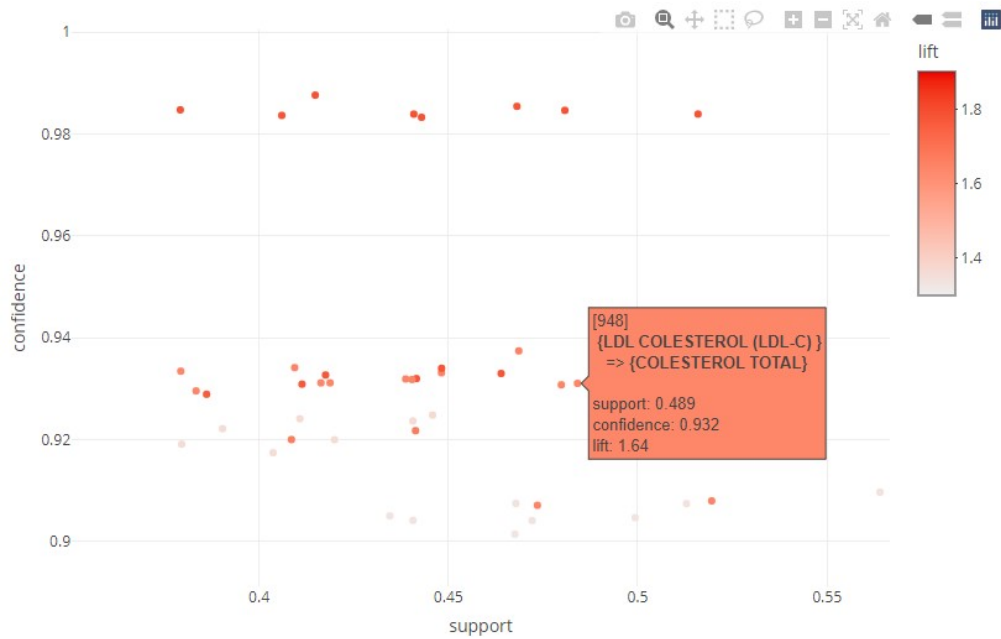
**Figura 3.** Distribuciones normalizadas con min-max de la cantidad de elementos por regla. Los datos mostrados corresponden a los algoritmos Apriori y Eclat ejecutados con soportes mínimos de 0,01 %, 1 % y 10 %.

Soporte mínimo	Cantidad de reglas
0,01 %	1144
0,1 %	973
0,5 %	558
1 %	450
5 %	146
10 %	146
20 %	66
30 %	66
50 %	7

**Cuadro 5.** Cantidad de reglas de asociación para cada uno de los Soportes mínimo configurados. Tanto Apriori como Eclat generan las mismas reglas de asociación para un mismo soporte mínimo.



**Figura 4.** Gráfico *Scatter Plot* para ver la cercanía de las reglas según sus valores para las medidas de interés.



**Figura 5.** Ejemplo de *zoom* y visualización de una regla en concreto.

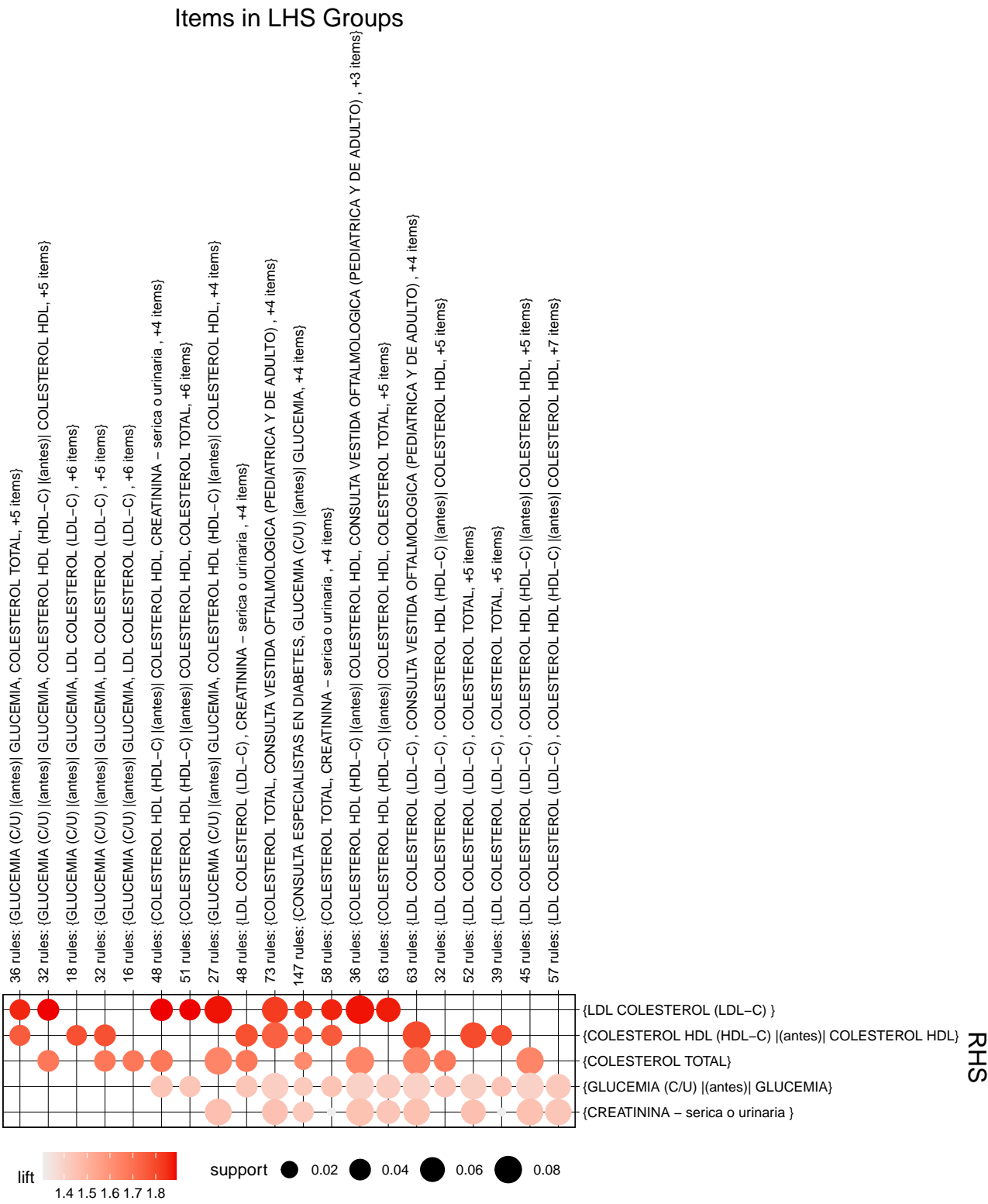
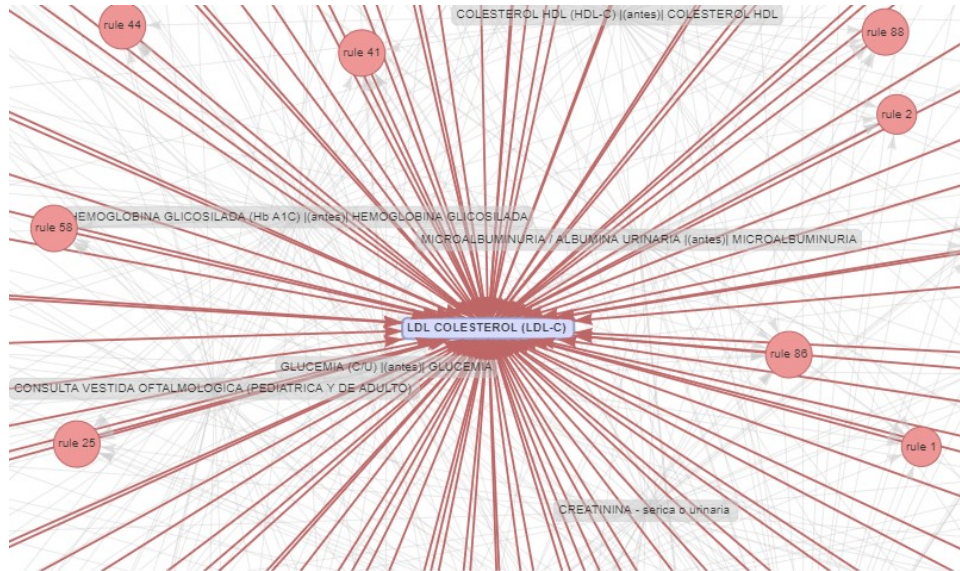
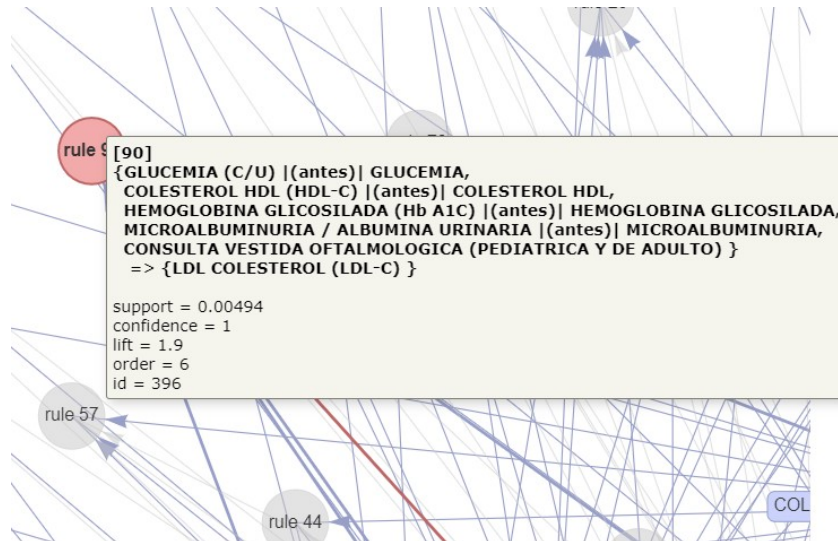


Figura 6. Gráfico *Grouped Matrix* para ver los diferentes antecedentes y consecuentes, junto con algunas medidas de interés.



**Figura 7.** Grafo: Un único ítem domina los consecuentes de las primeras 100 reglas ordenadas por *lift*.



**Figura 8.** Grafo: datos de interés de una regla particular.