UNIVERSIDAD DE SANTIAGO DE CHILE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA INFORMÁTICA

Laboratorio 3 - Reglas de asociación

Integrantes: Antonina Arriagada G.
Francisco Moreno L.

Curso: Inteligencia Computacional

Sección A-1

Profesor: Max Chacón Pacheco

8 de Junio de 2024

Tabla de contenidos

1. Introducción **1**

1.1. Objetivos 1

2. Marco teórico **2**

2.1. Reglas de asociación 2

2.1.1. Soporte 2

2.1.2. Confianza 3

2.2. Medidas de calidad 3

2.3. Monotonicidad 4

2.3.1. Otras propiedades de las medidas 5

3. Preprocesamiento **6**

3.1. Normalización de variables 6

3.2. Componentes principales (PCA) 6

4. Obtención de reglas **7**

4.0.1. Distribución de los datos por clase 7

4.1. Discretización de datos 9

4.2. Algoritmo Apriori 10

4.2.1. Despliegue de las mediciones 10

4.2.2. Filtro de reglas 11

5. Análisis de resultados **13**

5.1. Generación e inspección reglas CBA 13

5.2. Comparación de reglas 14

5.3. Clasificación mediante CBA 15

5.4. Cálculo de indicadores 16

5.5. Comparación entre distintos soportes para CBA 18

5.6. Comparación primera experiencia con actual 19

6. Conclusiones **21**

Bibliografía **23**

7. Anexos **24**

7.1. Algoritmo Apriori 24

7.1.1. Pasos del algoritmo 24

7.2. Algoritmo CBA 25

7.3. Figuras y Tablas 26

1. Introducción

En el presente informe se realizarán distintas implementaciones mediante el lenguaje de programación R a fin de poder generar reglas de asociación de variables con la búsqueda de alguna relación entre los objetos, lo cual conllevaría a una clasificación.

El contexto en cual se encuentra este laboratorio es una base de datos que representa simulaciones de edificaciones, las cuales son utilizadas para compararse entre sí, con el objetivo de investigar sobre las variables involucradas sobre la eficiencia energética. Con lo anterior, se determinan óptimos de carga de calefacción (HL) y carga de refrigeración (CL) en función de las dimensiones del edificio, el área acristalada y la orientación.

Para poder llevar acabo los procedimientos, se definen los siguientes objetivos:

1.1. Objetivos

1. Realizar un preprocesamiento de datos a fin de reducir dimensionalidad y preparar el conjunto de datos.

2. Utilizar un algoritmo de clasificación sobre el *dateset* a fin de abarcar reglas de asociación.

3. Analizar reglas interesantes a través de los resultados de soporte y confianza.

4. Comparar con conocimiento adquirido previamente.

2. Marco teórico

2.1. Reglas de asociación

Para poder encontrar relaciones interesantes y significativas entre un conjunto de datos, se definen las reglas de asociación como una implicación de tipo $X \implies Y$, donde X y Y son dos conjuntos disjuntos de datos. De lo anterior, se puede desprender la siguiente interpretación: "si el ítem X está presente en una transacción, entonces es probable que también se encuentre Y". Este tipo de reglas permiten mostrar la probabilidad entre relaciones de distintos elementos dentro de grandes conjuntos de datos, fue introducido por Agrawal et al. en 1993, tras la descripción de un gran conjunto de datos generado por la venta de supermercados. Con lo anterior, se identificaron en las transacciones qué productos llevaba un cliente. Un ejemplo práctico dado por la siguiente tabla de transacciones, con $X = \{\text{Pan}\}$ y $Y = \{\text{Leche}\}$:

Transacción	**Ítems**
1	Pan, Leche
2	Pan, Mantequilla
3	Leche, Huevos
4	Pan, Leche, Huevos
5	Pan, Leche

Cuadro 1: Transacciones de un minimarket

Una regla de asociación podría ser $\{\text{Pan}\} \rightarrow \{\text{Leche}\}$, lo que indica que los clientes que compran pan también tienden a comprar leche.

Además importante mencionar que el lado izquierdo de la regla recibe el nombre de antecedente o *left-hand-side* (LHS) y el lado derecho recibe el nombre de consecuente o *right-hand-side* (RHS).

2.1.1. Soporte

El soporte de una regla de tipo $\{\text{X}\} \rightarrow \{\text{Y}\}$, mide la proporción de transacciones que contienen ambos conjuntos X e Y. Se mide según:

$$\textit{support}(X \to Y) = \frac{\text{Número de transacciones que contienen } X \cup Y}{\text{Número total de transacciones}}$$

2.1.2. Confianza

La confianza de una regla mide la proporción de transacciones que contienen el conjunto Y entre aquellas que contienen X. Se define como:

$$\textit{confidence}(X \to Y) = \frac{\text{Número de transacciones que contienen } X \cup Y}{\text{Número de transacciones que contienen } X}$$

Una forma de interpretar la confianza consta de suponer la probabilidad de transaccionar un ítem Y si es que se lleva un ítem X.

2.2. Medidas de calidad

Las medidas de calidad son métricas que permiten cuantificar la eficacia de las reglas de asociación y evaluar la probabilidad de que reflejen relaciones significativas en los datos.

Lift: Evalúa la independencia entre X y Y. Es el cociente entre la confianza de la regla y la proporción de transacciones que contienen Y. Un valor de lift mayor que 1 indica que X y Y son positivamente correlacionados (ocurren más frecuentemente juntos de lo que se esperaría si fueran independientes) (Han et al., 2011):

$$\textit{lift}(X \to Y) = \frac{\text{confianza}(X \to Y)}{\text{soporte}(Y)}$$

Otra forma de verlo es:

$$\textit{lift}(X \to Y) = \frac{\text{soporte}(X \cup Y)}{\text{soporte}(X) \times \text{soporte}(Y)}$$

Cobertura (Coverage): Se define como el soporte del antecedente de la regla. Indica la frecuencia con la que el antecedente aparece en el conjunto de transacciones (Chacón, 2015):

$$\textit{coverage}(A \to B) = \text{soporte}(A)$$

Fisher Exact Test: Es una prueba estadística que calcula el p-valor asociado con la probabilidad de observar la regla solo por azar. Este test es útil para evaluar la significancia estadística de una regla de asociación, ayudando a determinar si la relación entre los ítems es significativa o podría haber ocurrido por azar (Fisher, 1922).

Leverage: Mide la diferencia entre la frecuencia observada de $X \cup Y$ y la frecuencia esperada si X y Y fueran independientes:

$$\mathit{leverage}(X \to Y) = \text{soporte}(X \cup Y) - \text{soporte}(X) \times \text{soporte}(Y)$$

Un valor positivo de leverage indica una relación positiva entre X y Y, mientras que un valor cercano a cero sugiere independencia.

Convicción (Conviction): Mide la dependencia de X y Y. Se define como la probabilidad de que X ocurra sin Y dividido por la probabilidad esperada de la ocurrencia de X sin Y:

$$\mathit{conviction}(X \to Y) = \frac{1 - \text{soporte}(Y)}{1 - \text{confianza}(X \to Y)}$$

Un valor de convicción mayor a 1 indica que X y Y están positivamente correlacionados.

Interest: Es la relación entre la frecuencia observada de X y Y y la frecuencia esperada si X y Y fueran independientes:

$$\mathit{interest}(X \to Y) = \frac{\text{soporte}(X \cup Y)}{\text{soporte}(X) \times \text{soporte}(Y)}$$

Similar al *lift*, pero se enfoca en la comparación directa de frecuencias observadas y esperadas.

2.3. Monotonicidad

La **especialización** de una regla de asociación ocurre cuando se agregan más condiciones al antecedente de la regla, haciendo que la regla sea más específica. Por el contrario, la **generalización** de una regla de asociación ocurre cuando se reducen las condiciones en el antecedente, haciendo que la regla sea más general (Chacón, 2015).

- **Especialización:** Si $A1 = V1 \wedge V2$ y $A2 = V1 \wedge V2 \wedge V3$, entonces $A2$ es una especialización de $A1$. Esto significa que $A2$ es más específica que $A1$.

- **Generalización:** Si $A2 = V1 \wedge V2 \wedge V3$ y $A1 = V1 \wedge V2$, entonces $A1$ es una generalización de $A2$. Esto significa que $A1$ es más general que $A2$.

Considerando lo anterior, la **monotonicidad** es una propiedad de las medidas de calidad de las reglas de asociación que describe cómo estas medidas cambian cuando se especializa una regla. Una medida es monótona si se comporta de manera predecible al agregar más condiciones a una regla (haciendo la regla más específica).

- Una medida es **monótona** si, al especializar una regla, la medida no disminuye. Formalmente, si $A1$ y $A2$ son dos especializaciones del antecedente tales que $|A1| < |A2|$, entonces la medida $\text{med}(A1)$ es menor o igual a la medida $\text{med}(A2)$:

$$\text{med}(A1) \leq \text{med}(A2)$$

- Una medida es **anti-monótona** si, al especializar una regla, la medida no aumenta:

$$\text{med}(A1) \geq \text{med}(A2)$$

2.3.1. Otras propiedades de las medidas

Transitividad: Algunas medidas pueden tener propiedades transitivas, lo que implica que si una regla $X \rightarrow Y$ y otra regla $Y \rightarrow Z$ tienen ciertas propiedades, se pueden inferir propiedades sobre la regla $X \rightarrow Z$ (Wikipedia contributors, 2024; Cuemath contributors, 2024).

Propiedad de No-negatividad: La mayoría de las medidas, como soporte, confianza, lift, leverage, y convicción, son no-negativas. Esto significa que sus valores no pueden ser negativos (Brin et al., 1997).

Interpretabilidad: Una buena medida de calidad debe ser interpretable, es decir, su significado y valor deben ser fácilmente comprensibles para los analistas de datos (Agrawal and Srikant, 1994).

Simetría: Algunas medidas pueden ser simétricas, es decir, la medida de $X \rightarrow Y$ es igual a la medida de $Y \rightarrow X$ (Pekkala, 2020).

Propiedad de Independencia: Algunas medidas pueden tener propiedades que indican independencia. Por ejemplo, el lift tiene un valor de 1 si los ítems son independientes (Fournier-Viger et al., 2012).

3. Preprocesamiento

Para poder trabajar con el *dataset* y con las reglas de asociación, primero es necesario realizar un preprocesamiento de los datos. En primera instancia, se revisó si era necesario eliminar algún dato o columna debido a la presencia de datos perdidos y de valores atípicos. Sin embargo, esto no fue necesario, ya que, tal como fue mencionado en el laboratorio 2, no existe la necesidad de realizar ninguna de estas dos acciones.

Por otra parte, se realizaron procesos de normalización, creación de datasets a partir del original y reducción de dimensionalidad a través de componentes principales, lo cual será explicado en las siguientes secciones y subsecciones.

Finalmente, aunque esto no será explicado en detalle en esta sección, ya que en parte tiene que ver con lo obtenido al inicio de la obtención de reglas, la última parte del preprocesamiento consistió en la fusión del *dataset* en el cual se realizó la reducción de dimensionalidad a través de PCA con el *dataset* donde se encuentran Y1 e Y2 discretizadas. Esto resultó en un nuevo *dataset* que combina la reducción de dimensionalidad con Y discretizadas.

3.1. Normalización de variables

Luego de realizar un análisis en el rango de los valores que pueden tomar las características, se determinó que estas tienen una diferencia muy significativa, por lo tanto, es necesario realizar una normalización de sus valores, para que las variables con valores muy grandes. Para esto se realizó una normalización utilizando la función de Python llamada *StandardScaler*, la cual estandariza las características restando la media y dividiendo por la desviación estándar, lo cual da como resultado una distribución con media 0 y desviación estándar 1. Se utiliza esta estandarización debido a que no asume ningún tipo de distribución específica y los datos, siendo no distribuidos normalmente, hacen que esta estandarización sea precisa para este caso.

3.2. Componentes principales (PCA)

Como última parte en el procesamiento de los datos, se decidió realizar un análisis de componentes principales con el objetivo de reducir la dimensionalidad del conjunto de

datos. Los resultados obtenidos se pueden observar en el laboratorio 1, donde, a través del método del codo, se determinó que el número óptimo de componentes era 5. Además, se realizó una revisión del porcentaje de la varianza explicada por cada componente, encontrando que con solo 5 componentes se puede explicar más del 90 % de la varianza. La diferencia entre el conjunto de datos final obtenido en el laboratorio 1 y el laboratorio 2 es que este último no será el *dataset* final utilizado, sino que solo una parte de él.

4. Obtención de reglas

Para poder empezar a trabajar con los datos, realizar los gráficos de distribución por clase y obtener reglas, fue necesario transformar las variables Y a variables discretizadas. En este caso, se decidió crear grupos que iban en rangos de 10 en 10, consiguiendo así crear 5 grupos para Y1. En cambio, para Y2 se crearon 4 grupos distintos. Estos grupos se pueden ver en la tabla 2. Las variables Y fueron discretizadas porque en este caso fueron tomadas como el target de las funciones de reglas.

Discretizacion	]-inf-10[	[10-20[	[20-30[	[30-40[	[40- inf[
Y1	HL1	HL2	HL3	HL4	HL5
Y2	-	CL1	CL2	CL3	CL4

Cuadro 2: discretizacion Y

Cabe destacar que el dataset obtenido mediante la reducción de dimensionalidad en el preprocesamiento, junto con el dataset que contiene las variables Y discretizadas, conforman el dataset final utilizado para comenzar a trabajar en la obtención de reglas.

4.0.1. Distribución de los datos por clase

Para ver la distribución de los datos por clases, se realizó una visualización de las variables X creadas por PCA junto con las salidas Y, utilizando diagramas de cajas, junto a un gráfico de densidad. Estos diagramas permiten visualizar la distribución de las variables en los conjuntos creados para cada salida. Los resultados obtenidos son los siguientes:

En lo que respecta a las variables X1 a X5 con respecto a Y1 se tiene lo siguiente:

La distribución de la variable X1, se puede ver que se encuentra fuertemente dividida entre dos valores, los menores a 1, los cuales corresponden a los grupos HL2 y HL1. Por otro lado, la otra parte de la distribución se encuentra cerca de valor 1 en adelante, formando parte del grupo de HL3 y HL4, esto se puede ver en la figura 1.

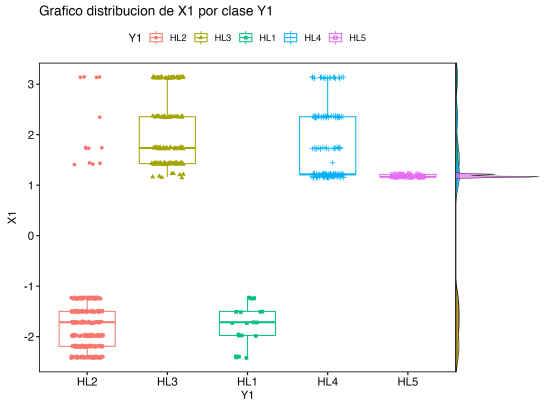

Figura 1: Diagrama de caja distribucion X1 en Y1

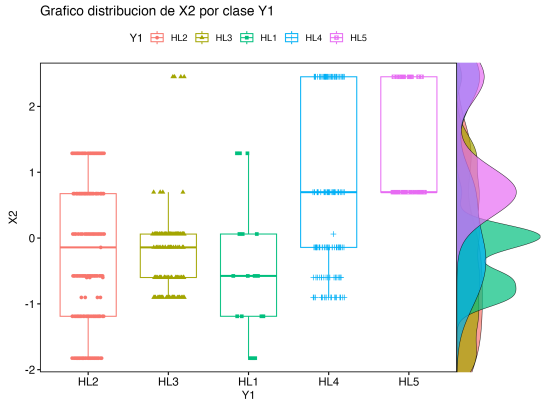

Figura 2: Diagrama de caja distribucion X2 en Y1

La distribución de la variable X2 en Y1, se puede ver que se encuentra dividida entre cinco valores. Siendo HL5 y HL3, los que se encuentran menos distribuidos en comparación a los otros tres grupos. Esto se puede ver en la figura 2 la cual se encuentra en el anexo.

Para X3 en Y1 ocurre algo muy parecido en cuanto a distribución con X2, siendo también 5 grupos, estos se pueden observar en la figura 8.

En cuanto a X4 y X5 en Y1, el primero de estos se ve dividido en 4 grupos o rangos de valores y para X5, es el que más dividido, se encuentra de estas 5 variables, siendo un total de 6 grupos, esto se puede ver en las figuras 9 y 10, las cuales se encuentran en el anexo.

Se realiza el mismo proceso para las variables X1 a X5 distribuidas en Y2. Las figuras obtenidas para estas variables son 11, 12, 13, 14 y 15, las cuales pueden observarse en el anexo.

Finalmente, a través de este proceso se pudo observar y determinar en qué grupo se distribuían mejor las variables. Por ejemplo, para la variable X5 en Y1, HL1 es el grupo que cuenta con la menor distribución de esta variable.

4.1. Discretización de datos

Para poder trabajar con los datos, fue necesario discretizarlos. Se utilizaron los resultados observados a través de la distribución de los datos por clase para definir los intervalos que se utilizarían para la discretización. Para esto, se decidió seguir como base la Escala de Likert, la cual permitió generar los intervalos. Esta variación dependió de la distribución de cada variable sobre las salidas Y1 e Y2, logrando crear escalas que van desde 2 puntos hasta algunas de 6 puntos.

Se puede observar la escala de 6 puntos usada para la variable X1 en Y1, desde la figura 3, esto es un ejemplo de todos los posibles valores utilizados en el resto de escalas de las variables.

Escala de Liker para X5 en Y1					
Muy bajo	Bajo	Moderadamente bajo	Moderadamente alto	Alto	Muy alto

Cuadro 3: Escala de liker para variable X5 en Y1

Para finalizar este punto, en las figuras 3 y 4 se puede observar un resumen de los valores obtenidos para cada variable con respecto a Y1 e Y2. En estas figuras se muestra qué escalas tienen y cuántos elementos de cada uno de los puntos de Likert.

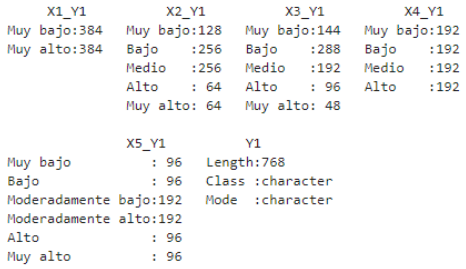

Figura 3: Resumen Likert para variables en Y1

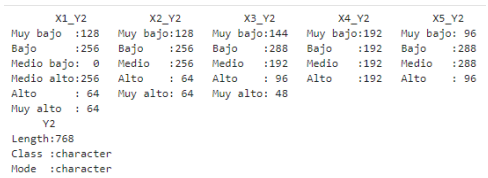

Figura 4: Resumen Likert para variables en Y2

Cabe destacar que, inicialmente, la discretización de estas variables se realizó de una manera completamente distinta. Debido a que los valores únicos de cada variable eran pequeños (ninguno superaba los 20 valores únicos), simplemente se habían discretizado uno por uno sin conformar grupos. Más adelante, esto conducía a obtener mejores resultados de

exactitud en las predicciones. Sin embargo, dado que esto solo era posible debido a la poca variación de valores, se decidió seguir una metodología más estándar.

4.2. Algoritmo Apriori

Con las operaciones anteriores podemos definir que se encuentra preparado el *dataset* para poder ser parámetro del algoritmo *Apriori*, el cual se encuentra definido en el anexo 7.1. Como otros parámetros de entrada, es necesario considerar un **soporte mínimo** y una **confianza mínima** para poder operar sobre el algoritmo. Para ello se utilizaron valores de soporte mínimo como un 5 % y valores de confianza mínima del 50 %. Con los valores por defecto del algoritmo (10 % soporte y 80 % de confianza) para Y2 (CL) se obtenían sólo 6 reglas. Se agrega además como parámetro una lista que permite establecerá los patrones a restringir dentro del espacio de búsqueda. Donde se definirán como consecuentes por separado tanto Y1 como Y2.

4.2.1. Despliegue de las mediciones

En el siguiente recuadro 4 se declara de la información obtenida por el algoritmo de manera *raw* o cruda.

Cuadro 4: Resumen de Medidas de calidad para Y1

	Support	Confidence	Coverage	Lift	Count
Min.	0.05078	0.5000	0.06250	1.019	39.00
1st Qu.	0.06250	0.5000	0.08333	1.019	48.00
Median	0.06380	0.9375	0.12500	1.931	49.00
Mean	0.09901	0.7856	0.13409	1.883	76.04
3rd Qu.	0.12174	1.0000	0.12500	2.037	93.50
Max.	0.47396	1.0000	0.50000	3.692	364.00

De lo anterior se obtienen 47 reglas que superan el índice de soporte y confianza mínimos. Donde tenemos 14 reglas de un ítem, 27 reglas de 2 ítems y 6 reglas de 3 ítems.

Por otro lado, para Y2 nos encontramos con el siguiente resumen 5:

Cuadro 5: Resumen de Medidas de calidad para Y2

	Support	Confidence	Coverage	Lift	Count
Min.	0.05208	0.5000	0.06250	1.103	40.00
1st Qu.	0.06250	0.7422	0.08333	2.207	48.00
Median	0.07161	1.0000	0.08333	2.207	55.00
Mean	0.09282	0.8652	0.11131	2.162	71.29
3rd Qu.	0.08333	1.0000	0.12500	2.207	64.00
Max.	0.33333	1.0000	0.33333	2.944	256.00

Donde tenemos 35 reglas en total, con 9 reglas de 1 ítem, 23 reglas formadas por dos ítems y 3 reglas formadas por 3 ítems.

4.2.2. Filtro de reglas

Se pueden observar mediantes los datos del recuadro 4 y 5, la combinación específica de datos y análisis estadísticos de las reglas obtenidas. No obstante, es esencial eliminar datos no significativos, redundantes y reglas contradictorias, asegurando al mismo tiempo una alta precisión y cobertura. Al emplear las funciones `duplicated()` e `is.redundant()` de R, se pueden eliminar los datos duplicados o repetidos dentro de las reglas generadas, resultando en los cuatros que se presentan a continuación.

Cuadro 6: Resumen de Medidas de calidad para Y1 (filtrada)

	Support	Confidence	Coverage	Lift	Count
Min.	0.05078	0.5000	0.06250	1.019	39.00
1st Qu.	0.06250	0.5000	0.08333	1.019	48.00
Median	0.06380	0.7656	0.12500	2.037	49.00
Mean	0.10538	0.7504	0.14953	1.894	80.94
3rd Qu.	0.12500	1.0000	0.17708	2.037	96.00
Max.	0.47396	1.0000	0.50000	3.692	364.00

En Y1 los resultados que podemos obtener tras el filtro, es una baja de 47 a 31 reglas. Donde de un ítem son 14, de dos ítems son 15 reglas y de tres ítems 2 reglas.

Cuadro 7: Resumen de Medidas de calidad para Y2 (filtrada)

	Support	Confidence	Coverage	Lift	Count
Min.	0.05208	0.5000	0.08333	1.103	40.00
1st Qu.	0.06348	0.5576	0.08333	1.844	48.75
Median	0.07096	0.6719	0.12500	2.122	54.50
Mean	0.11133	0.7180	0.15278	2.023	85.50
3rd Qu.	0.13151	0.8711	0.16667	2.211	101.00
Max.	0.33333	1.0000	0.33333	2.944	256.00

Con respecto a Y2 los resultados tras filtrar, determinan una baja de 35 a 12 reglas. Donde de un ítem son 9 y de dos ítems son 3 reglas.

En los siguiente gráficos se puede observar las reglas en el plano según confianza (*confidence*) en el eje y y soporte (*support*) en el eje x que superan el límite de 50 % establecido.

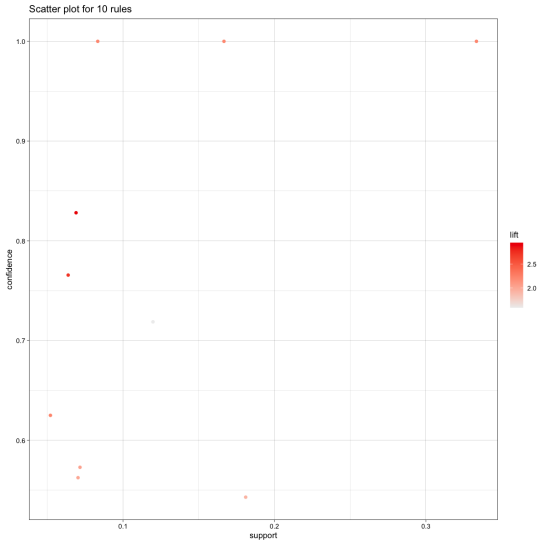

Figura 5: Reglas sobre min confianza Y1

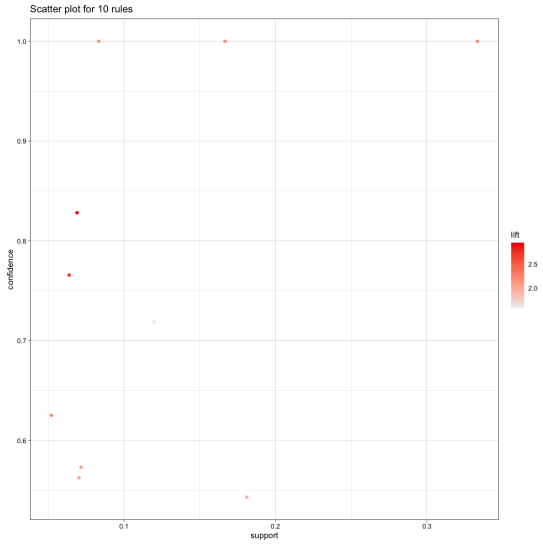

Figura 6: Reglas sobre min confianza Y2

5. Análisis de resultados

En esta sección se realizará el análisis de las reglas obtenidas anteriormente, teniendo en cuenta las medidas obtenidas por el algoritmo apriori y las pruebas con el clasificador CBA.

5.1. Generación e inspección reglas CBA

Al utilizar la función *CBA* de la librería *arulesCBA* (algoritmo CBA descrito en el anexo 7.2), con parámetros de confianza y soporte mínimo, los cuales equivalen al 5 % y 50 % respectivamente, para las variables X que tienen que ver con el target Y1, se obtiene un total de 16 reglas, de las cuales se pueden observar las primeras 2 en la tabla 8 , las cuales, en comparación a las 31 reglas obtenidas por el algoritmo *apriori*, es una baja considerable, lo cual significa que las reglas obtenidas en *CBA*, son más generales que las obtenidas anteriormente.

lhs	rhs	support	confidence	coverage	lift	count
X1_Y1=Muy bajo, X3_Y1=Bajo	Y1=HL2	0.1875	1	0.1875	2.03	144
X1_Y1=Muy bajo, X5_Y1=Mod. alto	Y1=HL2	0.125	1	0.125	2.03	96

Cuadro 8: CBA Y1

Por otro lado, para las variables con respecto a Y2, se obtuvieron 12 reglas en total. En comparación con las reglas obtenidas para Y1, el soporte más alto se encuentra en 0.33 y confidencia 1, mientras que el resto de las reglas se mantiene por debajo de 0.08, a excepción de un conjunto de reglas que se encuentra vacío con un soporte de 0.21 y confidencia 0.21. Este conjunto vacío equivale a las supuestas reglas relacionadas con CL2, lo cual indica que no se logró establecer una relación de reglas para este conjunto de valores de Y2. Esto puede significar una baja variabilidad de los datos o una falta de correlación. Siguiendo con las variables relacionadas a Y2, en el algoritmo apriori, son 12 reglas, igualando la cantidad de reglas obtenidas por ambos métodos, esto es posiblemente debido a los problemas mencionados anteriormente para obtener relación con el grupo CL2 de los datos.

5.2. Comparación de reglas

Al observar el cuadro 6 en relación con la tabla **??**, se puede notar más detenidamente la diferencia entre los valores máximos de soporte, junto con su *lift* y otros indicadores. En estos cuadros se observa una diferencia significativa en el soporte máximo y en el *lift* que estos valores generan. Esto se debe a que, al utilizar CBA, se generan reglas *podadas*, lo que reduce las repeticiones de estos elementos y permite un equilibrio.

Esto no significa que usar el algoritmo Apriori sea redundante; más bien, puede servir para identificar reglas interesantes que podrían haber sido podadas por CBA. Esto mismo se observa al comparar los resultados obtenidos con CBA para Y2 con los obtenidos con Apriori.

En este caso, en Y1, las reglas que más se repiten son las que tienen que ver con los valores de X1 entre]-inf y 1[y X3 entre]-1 y 0[, teniendo valores de muy bajo y bajo respectivamente, estos en relacion a HL2 de Y1.

Para Y2, el conjunto que claramente tiene más soporte, es la variable X1, en categoría bajo, la cual se encuentra en relación con el grupo CL1 de Y2.

A continuación se muestran de forma gráfica las reglas de asociación obtenidas por el algoritmo apriori, en como se relacionan las variables con las clases Y1 e Y2, la propiedad utilizada para asociar es el lift, lo cual ayuda a comprender de forma más gráfica que tan frecuente se describen las variables en Y1 o Y2. El color se oscurece mientras mayor sea el lift.

En la figura 7, se puede observar una primera parte del gráfico de reglas para Y1, en el cual se denota que el *lift* de estas es bajo debido a su tonalidad de color, en comparación a un *lift* alto que debería ser color rojo.

En el gráfico 16 la cual se encuentra en el anexo, se puede observar como las primeras tres reglas tienen un color rojo con tonalidad más fuerte, demostrando que estas tienen el *lift* mal alto entre las reglas, mostrando una clara relación con HL3 y HL4. Siendo la regla 1 (X1_Y1=Muy alto, X2_Y1=Bajo, X3_Y1=Medio) la que tiene relación con Y1=HL3, y las reglas 2 y 3 con relación a Y1 = HL4.

En cuanto a la figura 17, la cual se encuentra en el anexo, en esta se puede observar cómo las reglas 1 (*X2_Y2=Muy alto*) y 4 tienen un *lift* muy alto, ambas teniendo relación

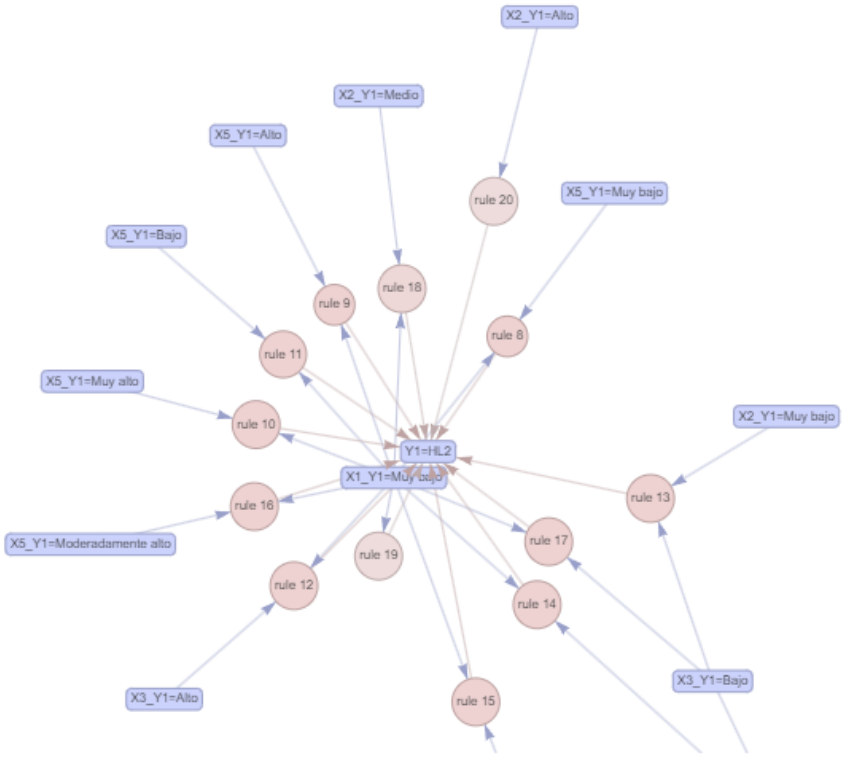

Figura 7: Gráfico reglas para Y1 parte 1

con $Y2 = CL3$.

Estos resultados muestran que los grupos de $Y1$ y $Y2$ son afectados por distintas reglas, permitiendo generar una clasificación basada en la frecuencia de aparición.

5.3. Clasificación mediante CBA

Un clasificador mediante CBA puede permitir establecer un acercamiento a qué clase debería pertecer un elemento en particular. En ese aspecto, hay que considerar que estamos subdiviendo HL (*Heating Load* como Y1) y CL (Cooling Load, como Y2), que son rangos de energía como resultado de distintas variables de la edificación. En ese mismo sentido, el clasificador puede también generar **falsos positivos**, los cuales son datos mal clasificados y por ende deberían estar catalogados en otra clase o rango.

En la diagonal principal de la matriz de confusión, se puede observar los **Verdaderos Positivos**, aquellos valores que fueron clasificados en el rango que le corresponde. En

Cuadro 9: Matriz de confusión para Y1

Clase	**HL1**	**HL2**	**HL3**	**HL4**	**HL5**
HL1	0	0	0	0	0
HL2	20	364	0	0	0
HL3	0	13	141	47	19
HL4	0	0	28	120	16
HL5	0	0	0	0	0

Cuadro 10: Matriz de confusión para Y2

Clase	**CL1**	**CL2**	**CL3**	**CL4**
CL1	348	36	0	0
CL2	0	52	20	12
CL3	0	77	196	27
CL4	0	0	0	0

los datos horizontales y verticales de la tabla podemos encontrar los **Falsos Positivos** en la diagonal superior y **Falsos Negativos** en la diagonal inferior.

Para obtener los **Verdaderos negativos** se debe identificar las celdas de la diagonal principal como verdaderos positivos y sumar las otras celdas, por ejemplo se puede ver en el cuadro 11 los valores de VN para ambas matrices según sus clases:

Cuadro 11: Verdaderos Negativos para Ambas Matrices

	HL1	**HL2**	**HL3**	**HL4**	**HL5**	**CL1**	**CL2**	**CL3**	**CL4**
VN	748	164	520	397	752	384	571	448	729

5.4. Cálculo de indicadores

Los siguientes indicadores que facilitan transparentar y analizar los resultados obtenidos de la matriz de confusión, donde la sensibilidad y la especificidad son métricas que estiman la capacidad del clasificador para discriminar casos positivos de los negativos. La sensibilidad se relaciona con los casos positivos y la especificidad los valores negativos.

$$\text{Sensibilidad} = \frac{VP}{VP + FN} \tag{1}$$

$$\text{Especificidad} = \frac{VN}{VN + FP} \tag{2}$$

La exactitud se declara como el grado de concordancia entre un valor verdadero y el medido.

$$\text{Exactitud} = \frac{VP + VN}{VP + VN + FP + FN} \tag{3}$$

Mientras que la precisión se define sobre lo cercano que está el resultado de la medición respecto al valor verdadero.

$$\text{Precisión} = \frac{VP}{VP + FP} \tag{4}$$

Establecidas las métricas se logró obtener lo siguiente para el algoritmo de CBA:

Clase	Sensibilidad	Especificidad	Precisión
HL1	NA	0.9739583	0.0000000
HL2	0.9479167	0.9661458	0.9655172
HL3	0.6409091	0.9489051	0.8343195
HL4	0.7317073	0.9221854	0.7185629
HL5	NA	0.9544271	0.0000000

Cuadro 12: Resultados de Sensibilidad, Especificidad y Precisión por Clase de Y1

Además, una exactitud general de 0.813802083333333 para Y1, mientras que para Y2 se logró un valor de 0.776041666666667, la cual está determinada por la cantidad de Valores Positivos en contraste al total de datos. Se puede hablar de sobre un 75 % de precisión del modelo.

Con respecto a la sensibilidad de HL1 y HL5, como también de CL4. Se debe analizar por qué no se está logrando obtener valores en la matriz de confusión respecto a estas clases. Se podría estimar que debido a que estas clases representan rangos límites de cargas de calor y de refrigeración, no existían tantos datos que lleguen a estas condiciones.

Clase	Sensibilidad	Especificidad	Precisión
CL1	0.9062500	1.0000000	1.0000000
CL2	0.6190476	0.8347953	0.3151515
CL3	0.6533333	0.9572650	0.9074074
CL4	NA	0.9492188	0.0000000

Cuadro 13: Resultados de Sensibilidad, Especificidad y Precisión por Clase de Y2

Estos márgenes en los rangos también se pueden evidenciar en las distribuciones por clase, donde los datos que están asociados a estos rangos se acumulan en ciertas valores y en otras se distribuyen a lo largo de eje y. Una posibilidad es a futuro no considerarlos y quitarlos de las clases estudiadas, en términos lógicos es extraño plantearse que una edificación requiera niveles muy bajos de capacidad calórica, mientras que si son muy altos puede referirse a otro tipo de edificaciones, como por ejemplo *malls* o megaconstrucciones.

5.5. Comparación entre distintos soportes para CBA

Para poder realizar una comparación con dos soportes se utilizó la librería `caret` la cual permite generar dos particiones de datos utilizando un 0.8 de los originales. Luego se generaron dos grupos, el primero fue ocupado de entrenamiento y el segundo de prueba. Esto permite realizar un contraste entre los clasificadores con distintos parámetros.

Esta operación fue realizada tanto para Y1 como para Y2, modificando sólo el valor del soporte y manteniendo el índice de confianza. Cuando se modifica el valor del soporte permitimos al algoritmo *apriori* realizar una poda distinta. Lo anterior quiere decir que si se disminuye el soporte mínimo, la poda se realizará con mayor libertad.

Una vez generado los modelos, nuevamente se establecieron matrices de confusión:

En Y1, el clasificador fuerte se obtuvo una precisión de 69,7 %, mientras que para el débil se obtuvo un valor de 70,3 %. Mientras que para Y2 el clasificador fuerte obtuvo una precisión de 61,8 %, mientras que el débil de 76,9 %.

Se puede establecer según lo anterior que si se aumenta el soporte mínimo el cálculo de clasificación del algoritmo CBA es menos preciso.

Lo anterior se puede analizar desde la perspectiva que al aumentar el soporte

Cuadro 14: Comparación de Modelos para Y1

Cuadro 15: Modelo soporte supp = 0.1

	HL1	**HL2**	**HL3**	**HL4**	**HL5**
HL1	0	0	0	0	0
HL2	4	73	0	0	0
HL3	0	2	33	33	7
HL4	0	0	0	0	0
HL5	0	0	0	0	0

Cuadro 16: Modelo soporte supp = 0.001

	HL1	**HL2**	**HL3**	**HL4**	**HL5**
HL1	4	0	0	0	0
HL2	0	73	1	0	0
HL3	0	2	30	13	2
HL4	0	0	2	20	0
HL5	0	0	0	0	5

Cuadro 17: Comparación de Modelos para Y2

Cuadro 18: Modelo soporte supp = 0.1

	CL1	**CL2**	**CL3**	**CL4**
CL1	51	0	0	0
CL2	0	0	0	0
CL3	18	33	43	7
CL4	0	0	0	0

Cuadro 19: Modelo soporte supp = 0.001

	CL1	**CL2**	**CL3**	**CL4**
CL1	66	0	0	0
CL2	3	22	14	6
CL3	0	11	29	1
CL4	0	0	0	0

mínimo en el algoritmo CBA puede llevar a reglas más generales, reduciendo así la precisión del clasificador, ya que estas reglas pueden no capturar patrones específicos de los datos. La especialización de reglas, por otro lado, puede reducir el soporte pero aumentar la confianza y precisión si las reglas son más descriptivas de los datos subyacentes. Por lo tanto, el objetivo que puede definirse es encontrar un balance adecuado en la selección de soporte mínimo para optimizar la precisión del algoritmo clasificador.

5.6. Comparación primera experiencia con actual

En la primera experiencia de laboratorio se realizó un análisis estadístico sólo se logró identificar que existían algunas variables que influían sobre Y1 e Y2 respectivamente. En esa misma línea se obtuvo una vista primitiva de cómo se comportaban los datos, por lo cual la experiencia actual descrita, permitió separar los valores de Y1 e Y2 en rangos, obteniendo nuevas visualizaciones. Se sigue manteniendo la relación de que las primeras cinco variables influyen fuertemente en los valores de salida, donde las reglas de asociación nos permiten

establecer definiciones más certeras.

Por otra parte, al comparar los resultados obtenidos al realizar el análisis de componentes principales, los cuales indicaban qué variable de las cinco que quedaron después de la reducción, se relaciona con cada una de las características, se observan fuertes relaciones con las reglas obtenidas. Por ejemplo, en el laboratorio 1, se llegó a la conclusión de que el componente 4 es el único que incluye la característica X6, la cual, a través de métodos de correlación entre otros, se concluyó que no tiene mucha relación con ninguna de las dos salidas Y1 e Y2. Esto se demuestra ya que el componente principal 4 o X4, no aparece en ninguna de las reglas creadas por CBA, lo que significa que está en las reglas vacías, ya que no se logró encontrar una relación significativa entre estas.

6. Conclusiones

A lo largo de esta investigación, se lograron alcanzar varios objetivos. En primer lugar, se logró la creación y desarrollo de un programa en R, el cual permitió implementar los algoritmos de clasificación por reglas de asociación sobre el *dataset*. Para esto, fue necesario realizar un preprocesado correspondiente a la dimensionalidad de los datos, con el fin de obtener los mejores resultados posibles.

Además, se llevó a cabo un análisis de los resultados obtenidos, comparándolos con la literatura existente. Esta comparación permitió mejorar el proceso de análisis y aprendizaje, ya que la información obtenida de la literatura fue utilizada para optimizar el análisis.

Por otra parte, a través de la obtención de reglas, se logró analizar una serie de reglas interesantes mediante los resultados de soporte y confianza.

Finalmente, en cuanto al último objetivo planteado, se logró comparar los resultados obtenidos con el conocimiento obtenido anteriormente en el laboratorio 1.

En el desarrollo de la experiencia, se lograron obtener múltiples resultados. En el preprocesamiento, se observó y concluyó, al igual que en la experiencia del laboratorio 2, que solo era necesario realizar una normalización debido a la amplia diferencia de rangos, y una disminución de dimensionalidad a través de PCA, un paso que la mayoría de los conjuntos de datos necesitan.

Estas condiciones permitieron mantener la integridad de los datos utilizados al máximo posible, facilitando un análisis más eficiente y preciso.

Por otra parte, al realizar el trabajo previo para la obtención de reglas, se observó que debido a la pequeña cantidad de valores únicos del *dataset* utilizado, era posible discretizar cada valor sin necesidad de agrupar, obteniendo resultados mucho más certeros. Sin embargo, como esto solo era posible debido al tamaño de los datos únicos, se tomó la ruta más general, realizando todo el proceso de agrupación y discretización de datos. El plan original era realizar todo el trabajo en Python, pero debido a problemas técnicos con la función *pyArc*, esto no fue posible y se tuvo que utilizar R para obtener los mejores resultados posibles. Aunque esto complicó el trabajo, permitió obtener resultados aún mejores de los

previstos inicialmente, aunque con una gran pérdida de tiempo.

En cuanto a las reglas obtenidas, se pudo observar una clara diferencia entre las reglas obtenidas para el grupo de la clase Y1 y la clase Y2, mostrando las claras diferencias en las necesidades de distintas características para las salidas esperadas. En comparación con otros trabajos en los que solo se necesita realizar el análisis y obtención de reglas, en este caso fue necesario crear dos *datasets* distintos, uno por cada clase (objetivo). Esto, como se mencionó anteriormente, sirvió aún más para comparar cómo estas variaban a través de los distintos algoritmos, en este caso, Apriori y CBA.

Por otra parte, al comparar la *accuracy* o exactitud obtenida de las reglas para Y1 e Y2, se puede observar una diferencia del 4 %. Esto podría deberse a que las características afectaban de forma más directa a Y1, o que debido a la reducción de dimensionalidad, ese 10 % perdido al ser realizada afectó más los resultados.

Finalmente, se puede decir que todo el trabajo realizado en el laboratorio 1 puede explicar en gran parte los resultados obtenidos en esta experiencia, lo que demuestra cómo los elementos trabajados están relacionados con los futuros trabajos, en este caso, las reglas de asociación.

Bibliografía

Agrawal, R., Imieliński, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 207–216. ACM.

Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules. In *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, pages 487–499.

Brin, S., Motwani, R., Ullman, J. D., and Tsur, S. (1997). Dynamic itemset counting and implication rules for market basket data. *ACM SIGMOD Record*, 26(2):255–264.

Chacón, M. (2015). *Minería de Datos, Capítulo VI: Reglas de Asociación*. Material distribuido por profesor.

Cuemath contributors (2024). Transitive relations - definition, examples, properties. *Cuemath*.

Fisher, R. A. (1922). On the interpretation of χ^2 from contingency tables, and the calculation of p. *Journal of the Royal Statistical Society*, 85(1):87–94.

Fournier-Viger, P., Gomariz, A., Campos, R., and Thomas, R. (2012). A survey of pattern mining. *Springer*, pages 1–31.

Han, J., Kamber, M., and Pei, J. (2011). *Data Mining: Concepts and Techniques*. Elsevier, Waltham, MA, 3rd edition.

Pekkala, K. (2020). Understanding symmetry in association rule mining. *Journal of Data Science*, 18(1):123–135.

Wikipedia contributors (2024). Transitive relation. *Wikipedia, The Free Encyclopedia*.

7. Anexos

7.1. Algoritmo Apriori

Método de la minería de datos que posibilita encontrar patrones en grandes bases de datos a fin de generar reglas de asociación. La idea principal está representada por encontrar conjuntos de datos (*itemsets*) según las frecuencia que tengan en transacciones que se identifiquen. Para poder llevar acabo este algoritmo se requiere realizar lo siguiente:

Generación de itemsets frecuentes: En esta fase, el algoritmo identifica todos los *itemsets* que cumplen con un umbral mínimo de soporte, lo que quiere decir, que aparecen en al menos un cierto porcentaje de las transacciones.

Generación de reglas de asociación: Según los *itemsets* frecuentes, el algoritmo genera reglas de asociación que cumplen con un umbral mínimo de confianza, indicando la probabilidad de que un *itemset* ocurra dado que otro *itemset* ya ha ocurrido.

7.1.1. Pasos del algoritmo

Algorithm 1 Algoritmo Apriori

Input: Base de datos de transacciones D, umbral mínimo de soporte min_sup
Output: Conjuntos de ítems frecuentes L
$L_1 \leftarrow \{\text{itemsets de 1 ítem con } min\_sup\}$
$k \leftarrow 2$
while $L_{k-1} \neq \emptyset$ **do**
    $C_k \leftarrow$ generar candidatos de L_{k-1}
    for each transacción $t \in D$ **do**
        incrementar el conteo de los candidatos en C_k que están contenidos en t
    end for
    $L_k \leftarrow \{c \in C_k \mid c.support \geq min\_sup\}$
    $k \leftarrow k + 1$
end while
return $\bigcup_i L_i$

7.2. Algoritmo CBA

Por otro lado, se tiene el algoritmo CBA (*Classification Based on Associations* en inglés), que considera no sólo las reglas de asociación si no que en base a estas adhiere una componente de clasificación. Este clasificador posibilita la realización de predicciones basadas en las relaciones entre los atributos de los datos y la clase objetivo.

Algorithm 2 Algoritmo CBA (Classification Based on Associations)

Input: Base de datos de transacciones D, umbral mínimo de soporte min_sup, umbral mínimo de confianza min_conf
Output: Clasificador basado en reglas de asociación
Fase 1: Generación de Reglas de Asociación
$AR \leftarrow$ Aplicar algoritmo de minería de reglas de asociación (e.g., Apriori) a D
$AR \leftarrow \{r \in AR \mid r.support \geq min\_sup \text{ y } r.confidence \geq min\_conf\}$
Fase 2: Filtrado y Ordenamiento de Reglas
Ordenar AR en función de confianza, soporte, y otras métricas relevantes
Fase 3: Construcción del Clasificador
$Classifier \leftarrow \emptyset$
for each regla $r \in AR$ **do**
    $Classifier \leftarrow Classifier \cup \{r\}$
end for
Fase 4: Clasificación de Nuevos Ejemplos
function CLASSIFY($example$)
    for each regla $r \in Classifier$ **do**
        if $r.antecedent \subseteq example$ **then**
            return $r.consequent$
        end if
    end for
    return clase por defecto
end function

7.3. Figuras y Tablas

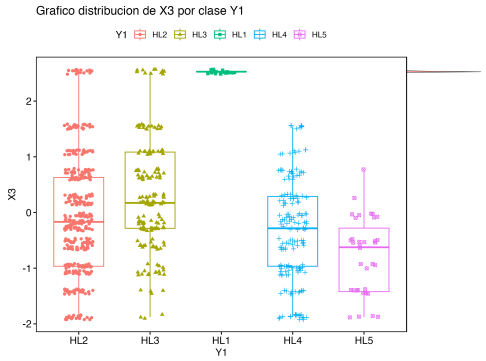

Figura 8: Diagrama de caja distribucion X3 en Y1

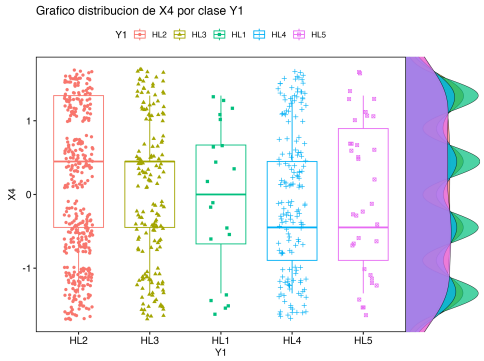

Figura 9: Diagrama de caja distribucion X4 en Y1

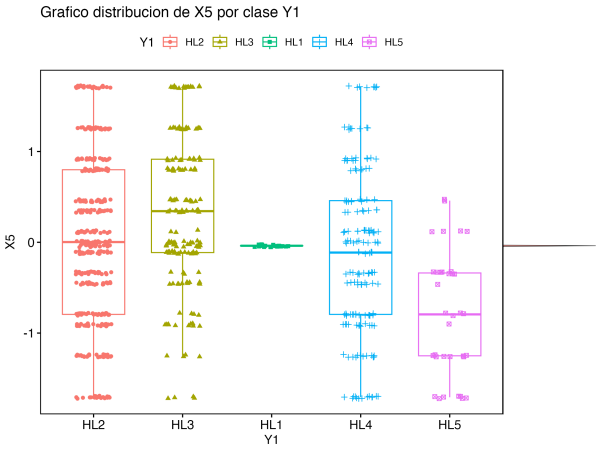

Figura 10: Diagrama de caja distribucion X4 en Y1

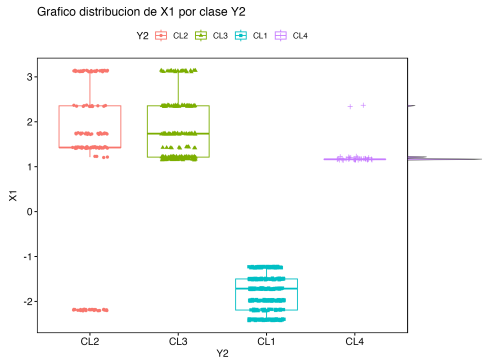

Figura 11: Diagrama de caja distribucion X1 en Y2

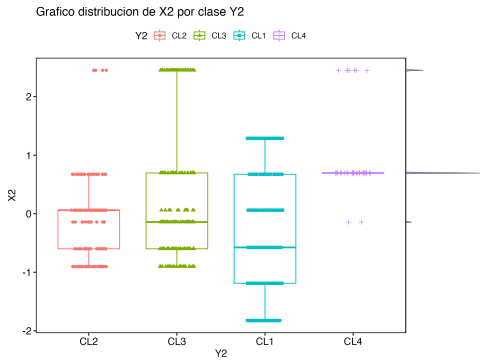

Figura 12: Diagrama de caja distribucion X2 en Y2

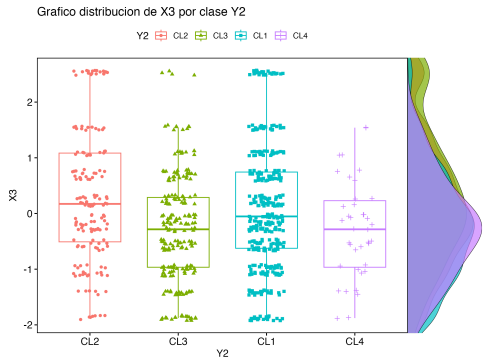

Figura 13: Diagrama de caja distribucion X3 en Y2

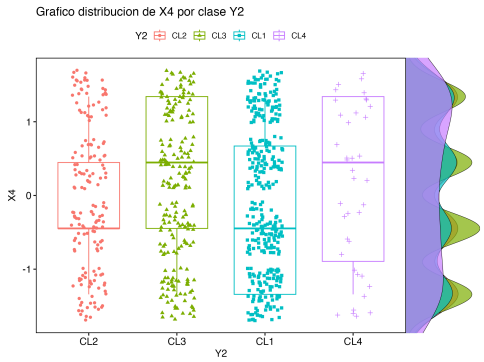

Figura 14: Diagrama de caja distribucion X4 en Y2

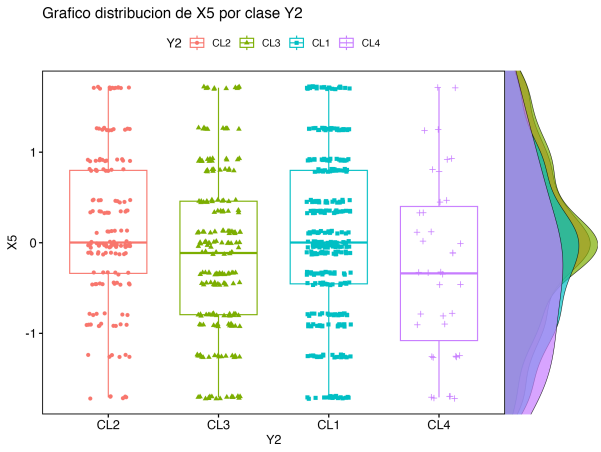

Figura 15: Diagrama de caja distribucion X5 en Y2

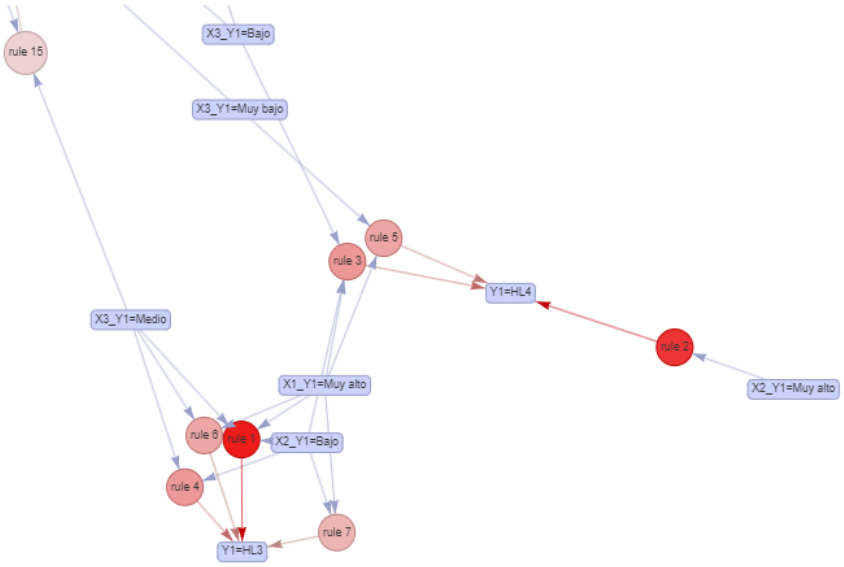

Figura 16: Gráfico reglas para Y1 parte 1

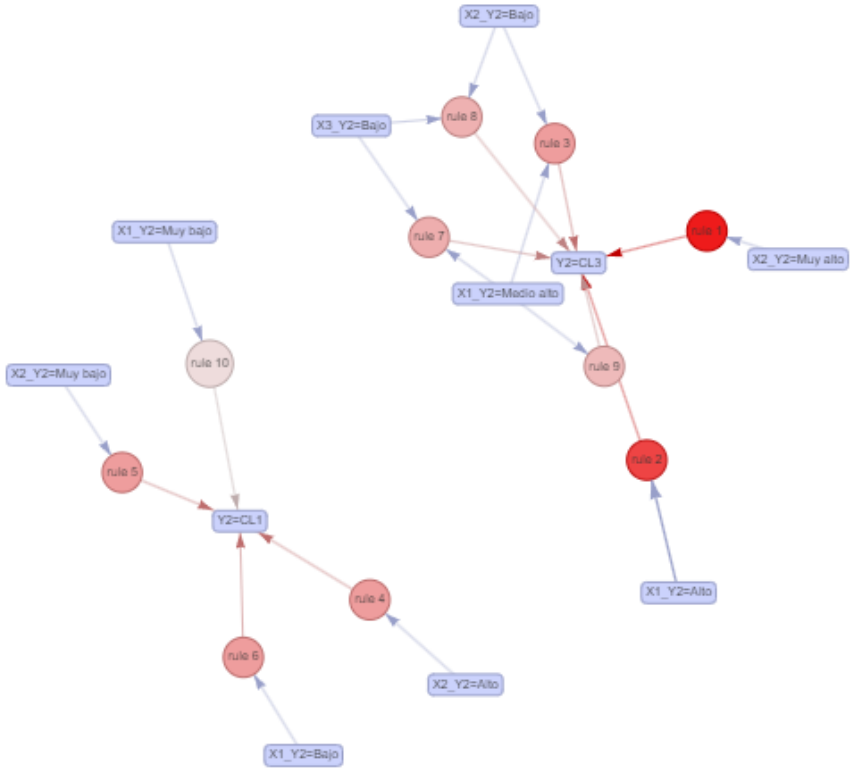

Figura 17: Gráfico reglas para Y2