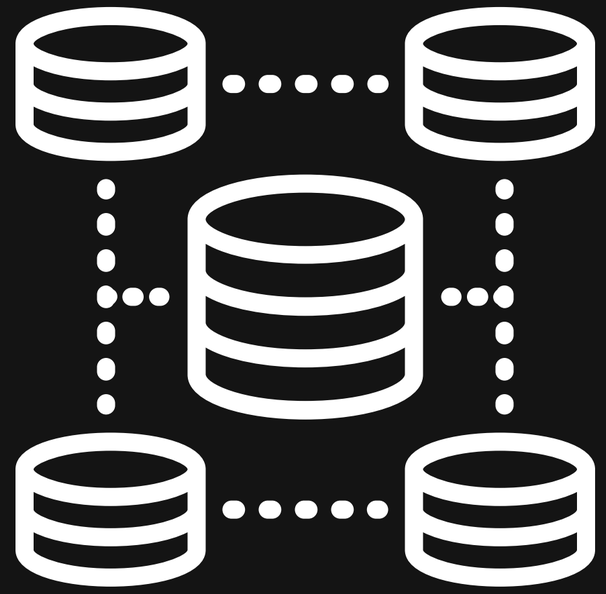
Reglas de asociación

Equipo 3:

- Ruy Aramis López Verduzco 1863861
- Ricardo González Berumen 1941497
- Thalia Ruiz Espitia 1941494
- José Claudio Gaytán Guitiérrez 1855455
- Nancy Janeth Rodríguez Pacheco 1845816
- Daisy Aideth Gonzalez Martinez

¿Qué son las reglas de asociación?

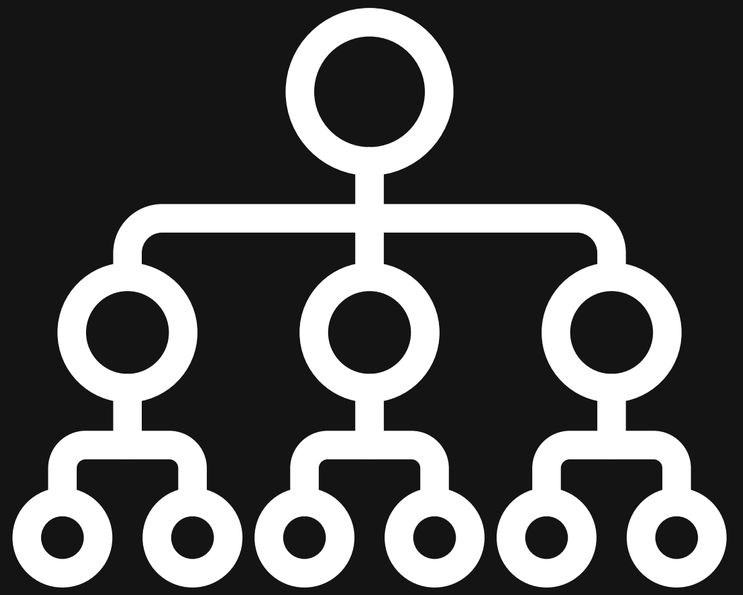
Búsqueda de patrones frecuentes, asociaciones, correlaciones o estructuras causales entre conjuntos de elementos u objetos en bases de datos de transacciones, bases de datos relacionales y otros repositorios de información disponibles.



¿Qué son las reglas de asociación?

A cada uno de los eventos o elementos que forman parte de una transacción se le conoce como item y a un conjunto de ellos itemset. Una transacción puede estar formada por uno o varios items, en el caso de ser varios, cada posible subconjunto de ellos es un itemset distinto.

Por ejemplo, la transacción T = {A,B,C} está formada por 3 items (A, B y C) y sus posibles itemsets son: {A,B,C}, {A,B}, {B,C}, {A,C}, {A}, {B} y {C}.



¿Cómo funcionan?

Una regla de asociación tiene dos partes:

- un antecedente (if) y un consecuente (then)

Un antecedente es un elemento que se encuentra dentro de los datos.
Un consecuente es un elemento que se encuentra en combinación con el antecedente.

¿Cómo funcionan?

Las reglas de asociación se crean buscando en los datos patrones frecuentes de “if-then” y utilizando los criterios de soporte y confianza para identificar las relaciones más importantes. El soporte del item o itemset(X) es el número de transacciones que contienen X dividido entre el total de transacciones. La confianza mide que tan frecuente un itemset aparece en transacciones que contienen X.

Objetivo

Dado un conjunto de transacciones T, el objetivo de la minería de reglas de asociación es encontrar todas las reglas teniendo:

- Umbral mínimo de soporte
- Umbral mínimo de confianza

Si los itemset no superan el mínimo de confianza son descartados, de igual manera sucede con la confianza.

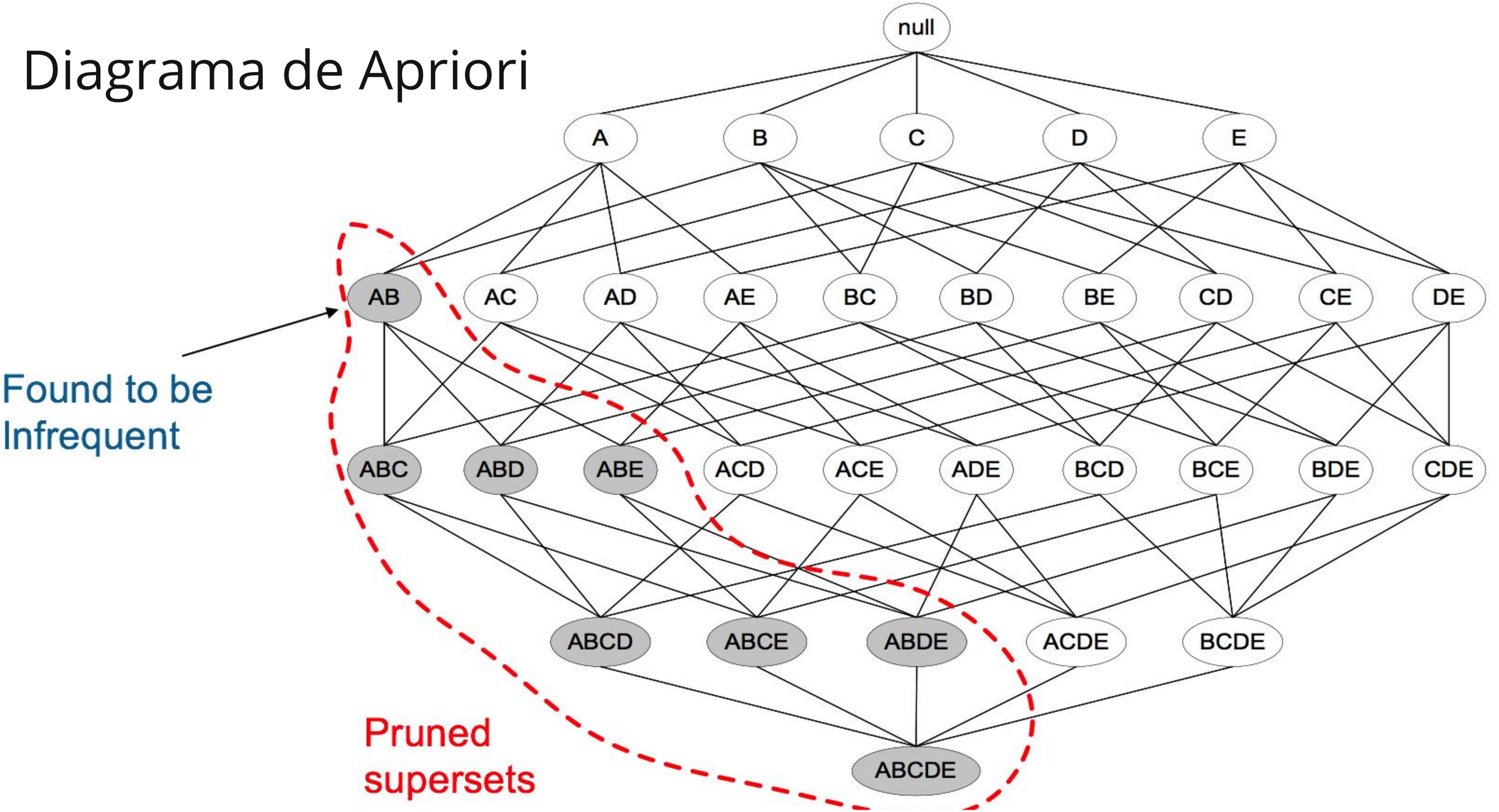
Algoritmos de reglas de asociación

Apriori

Apriori fue uno de los primeros algoritmos desarrollados para la búsqueda de reglas de asociación y sigue siendo uno de los más empleados, tiene dos etapas:

- Identificar todos los itemsets que ocurren con una frecuencia por encima de un determinado límite (itemsets frecuentes).
- Convertir esos itemsets frecuentes en reglas de asociación.

Diagrama de Apriori

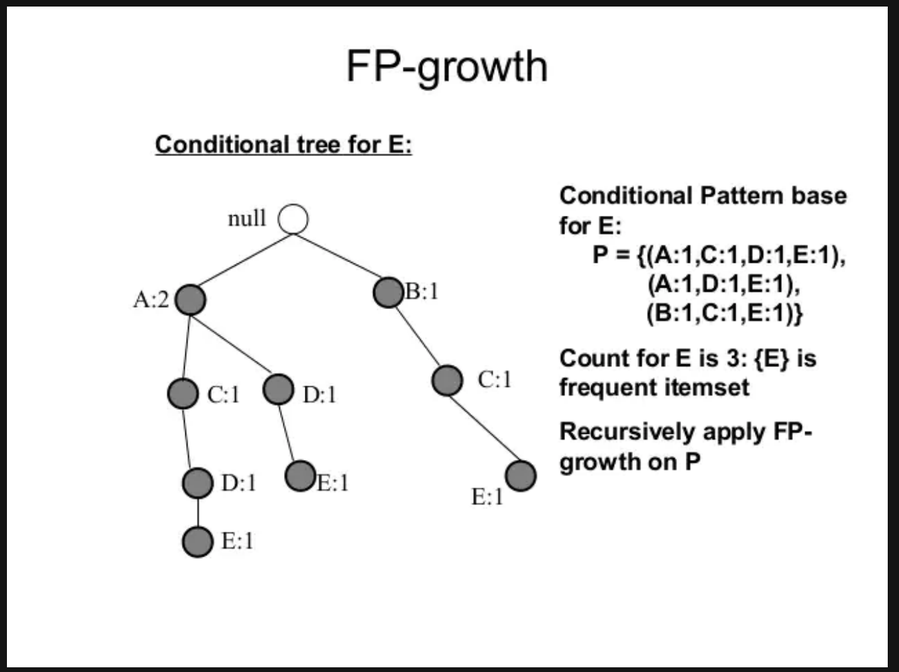


Algoritmos de reglas de asociación

FP-Growth

FP-Growth permite extraer reglas de asociación a partir de itemsets frecuentes pero, a diferencia del algoritmo Apriori, estos se identifican sin necesidad de generar candidatos para cada tamaño.

En términos generales, el algoritmo emplea una estructura de árbol (Frequent Pattern Tree) donde almacena toda la información de las transacciones.



Algoritmos de reglas de asociación

Eclat

La principal diferencia entre este algoritmo y Apriori es la forma en que se escanean y analizan los datos. El algoritmo Apriori emplea transacciones almacenadas de forma horizontal, es decir, todos los elementos que forman una misma transacción están en la misma línea. El algoritmo Eclat, sin embargo, analiza las transacciones en formato vertical, donde cada línea contiene un item y las transacciones en las que aparece ese item.

¿Dónde se utilizan las reglas de asociación?

MEDICINA

Los médicos pueden utilizar las reglas de asociación para diagnosticar pacientes. Mediante el uso de análisis de datos, los médicos pueden determinar la probabilidad condicional de una enfermedad comparando las relaciones de los síntomas en los datos de casos anteriores

VENTA AL POR MENOR

Los minoristas pueden recopilar datos sobre los patrones de compra, registrando los datos de las compras a medida que los códigos de barras de los artículos son escaneados.

ENTRETENIMIENTO

Servicios como Netflix y Spotify pueden utilizar las reglas de asociación para alimentar sus motores de recomendación de contenido.

NETFLIX

prime video

Ejemplo

Utilizamos una base de datos de transacciones de una tienda para realizar el ejemplo, específicamente esta: https://www.kaggle.com/heeraldedhia/groceries-dataset

Utilizamos el algoritmo apriori, ya que R cuenta con una libreria llamada "arules" con la función de apriori. También se utiliza, en menor medida, la libreria "tidyverse"

Primeramente, se extraen y leen los datos de la base de datos de la tienda

```r
#Lectura de datos
transacciones <- read.transactions(file = "C:/Users/Ruy Aramis/Downloads/Groceries_dataset1.csv",
                                   format = "single", sep = ",", header = T,
                                   cols = c("Member_number", "itemDescription"),
                                   rm.duplicates = TRUE)
transacciones
```

```
> transacciones
transactions in sparse format with
 3898 transactions (rows) and
 167 items (columns)
```

La base de datos consta de 3898 transacciones con 167 items,
que en este caso serían los productos de la tienda

Al manipular los datos, podemos observar las transacciones #6 y #7.
También vemos la transacción con el menor número de items, la que tiene el mayor, media, etc

```
> inspect(transacciones[6:7])
    items                    transactionID
[1] {margarine,
     rolls/buns,
     whipped/sour cream}          1005
[2] {bottled beer,
     bottled water,
     chicken,
     chocolate,
     flour,
     frankfurter,
     rice,
     rolls/buns,
     shopping bags,
     skin care,
     softener,
     whole milk}                  1006
> summary(size(transacciones))
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   6.000   8.500   8.919  12.000  26.000
```

```r
#Itemsets
soporte <- 20/dim(transacciones)[1]
itemsets <- apriori(data = transacciones,
                    parameter = list(support = soporte, confidence = 0.70,
                                     minlen = 1, maxlen = 15, target = "frequent itemset"))
summary(itemsets)
```

```
> summary(itemsets)
set of 9876 itemsets

most frequent items:
      whole milk other vegetables       rolls/buns             soda           yogurt          (Other)
            2927             2345             2073             1729             1654            19410

element (itemset/transaction) length distribution:sizes
   1    2    3    4    5    6
 141 2099 4975 2432  228    1
```

Al hacer el summary, podemos ver que hay 9,876 itemsets que cumplen con el mínimo de soporte, y la mayoría estan formados por 3 items

Ahora se ven los 10 itemset con mejor soporte

```
> inspect(top10.item)
      items                          support   transIdenticalToItemsets count
[1]   {whole milk}                   0.4581837 0.0002565418             1786
[2]   {other vegetables}             0.3766034 0.0005130836             1468
[3]   {rolls/buns}                   0.3496665 0.0002565418             1363
[4]   {soda}                         0.3134941 0.0002565418             1222
[5]   {yogurt}                       0.2829656 0.0000000000             1103
[6]   {tropical fruit}               0.2337096 0.0002565418              911
[7]   {root vegetables}              0.2306311 0.0000000000              899
[8]   {bottled water}                0.2136993 0.0000000000              833
[9]   {sausage}                      0.2060031 0.0000000000              803
[10]  {other vegetables,whole milk}  0.1913802 0.0010261673              746
```

El item de leche entera es el que tiene mejor soporte y en el top 10 se encuentra un itemset de 2 items, que a su vez contiene a los 2 items con mayor soporte

Como se observa, diferencia principal para encontrar el soporte y la confianza es el parametro target, en el primero se utiliza "frequent itemset" y en este "rules", ya que no solo encuentra el soporte y la confianza, sino también la cobertura y el lift

```
#Reglas de asociación
reglas <- apriori(data = transacciones,
                  parameter = list(support = soporte, confidence = 0.805, target = "rules"))
summary(reglas)
```

```
> summary(reglas)
set of 25 rules

rule length distribution (lhs + rhs):sizes
 3  4  5
 2 14  9
```

Entonces, hay 25 itemsets que cumplen con mínimo requerido de confianza, donde la mayoría consta de 4 items

Así como se hizo con el soporte, se busca el top 10 de itemsets con mayor confianza

```
> top10 <- sort(reglas, by = "confidence", decreasing = TRUE)[1:10]
> inspect(top10)
     lhs                                                rhs          support     confidence
[1]  {ham,pastry,yogurt}                             => {whole milk} 0.005387378 0.9545455
[2]  {domestic eggs,meat,other vegetables}           => {whole milk} 0.005643920 0.8800000
[3]  {brown bread,meat,other vegetables}             => {whole milk} 0.005130836 0.8695652
[4]  {bottled water,pip fruit,rolls/buns,yogurt}     => {whole milk} 0.005130836 0.8695652
[5]  {brown bread,curd,soda}                         => {whole milk} 0.006670087 0.8666667
[6]  {brown bread,rolls/buns,shopping bags,yogurt}   => {whole milk} 0.005900462 0.8518519
[7]  {brown bread,canned beer,curd}                  => {whole milk} 0.005387378 0.8400000
[8]  {brown bread,rolls/buns,shopping bags,soda}     => {whole milk} 0.005387378 0.8400000
[9]  {bottled water,rolls/buns,root vegetables,yogurt} => {whole milk} 0.006670087 0.8387097
[10] {ham,rolls/buns,root vegetables}                => {whole milk} 0.006413545 0.8333333
```

lhs (left-hand-side) sería el "**if** " y rhs (right-hand-side) el "**then**". De acuerdo con esto, la tabla nos diría que cada que se compra jamón, pan dulce y yogurt, al 95% de confianza, también se compra leche

5 preguntas

¿Qué tan frecuente las vemos en nuestro día a día?

¿Qué ventajas nos ofrecen?

¿Que tanta influencia tiene el machine learning en estas reglas?

¿Qué tan relacionado está con el Big Data?

¿Qué algoritmo es más utilizado?

Referencias

- https://www.cienciadedatos.net/documentos/43_reglas_de_asociacion#Introducci%C3%B3n
- https://aprendeia.com/reglas-de-asociacion/
- https://blog.jaywrkr.com/miner%C3%ADa-de-datos-3-f75d15f90c46
- http://datascience.esy.es/wiki/reglas-de-asociacion/
- https://blogs.imf-formacion.com/blog/tecnologia/reglas-asociacion-big-data-202007/