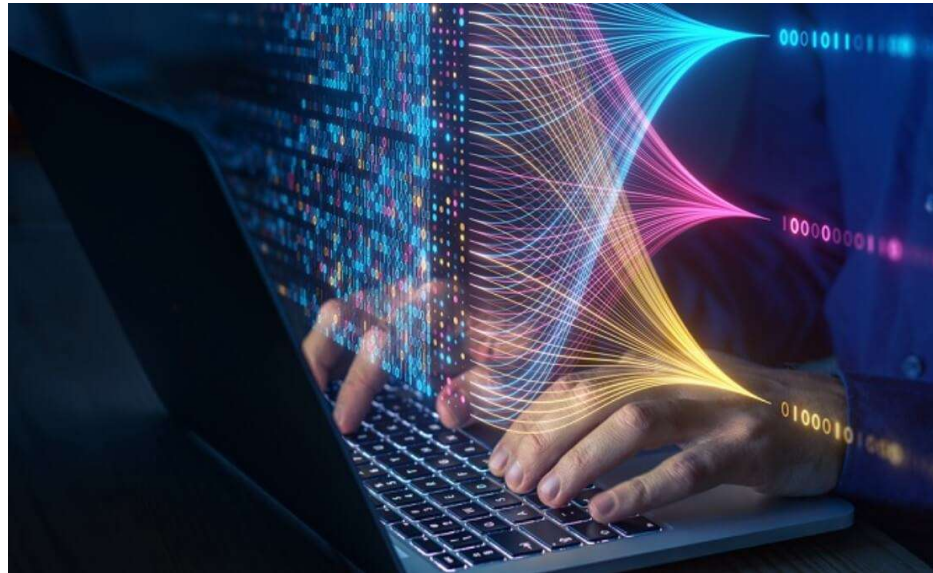


# Técnicas de inteligencia artificial

Adriana Cervantes Castillo



## SEMANA 5: REGLAS

# Contenido

- Sistemas basados en reglas-características
- Reglas de clasificación y reglas de asociación

# Sistemas basados en reglas

Los sistemas de reglas son uno de los métodos más extendidos para representar conocimiento. Algunas ventajas de los sistemas de reglas:



La sintaxis básica de una regla es:

SI <antecedente>

ENTONCES <consecuente>

- Las reglas pueden formarse por una o más conjunciones o disyunciones. No es recomendable mezclar conjunciones y disyunciones en una sola regla.

SI <antecedente 1>

AND <antecedente 2>

AND <antecedente 3>

ENTONCES <consecuente 1>

<consecuente 2>

```
SI edad < 25
    AND "años con carné de conducir" < 2
    AND "número de siniestros previos" > 0
ENTONCES "riesgo de siniestro" es alto
```

- Son una alternativa a los árboles de decisión.
- El antecedente → valores de los atributos
- El consecuente → valor de la clase.

# Reglas de clasificación y de asociación

- Las reglas de clasificación predicen la clase,
- Las reglas de asociación predicen valores de atributos, combinaciones de valores de atributos, o la propia clase.
- El interés de las reglas de asociación es descubrir combinaciones de pares atributo-valor que ocurren con frecuencia en un conjunto de datos.

Id	Ambiente	Temperatura	Humedad	Viento	Jugar
E1	soleado	Alta	Alta	Falso	No
E2	soleado	Alta	Alta	Verdadero	No
E3	nublado	Alta	Alta	Falso	Sí
E4	Lluvioso	Media	Alta	Falso	Sí
E5	Lluvioso	Baja	Normal	Falso	Sí
E6	Lluvioso	Baja	Normal	Verdadero	No
E7	Nublado	Baja	Normal	Verdadero	Sí
E8	Soleado	Media	Alta	Falso	No
E9	Soleado	Baja	Normal	Falso	Sí
E10	Lluvioso	Media	Normal	Falso	Sí
E11	Soleado	Media	Normal	Verdadero	Sí
E12	Nublado	Media	Alta	Verdadero	Sí
E13	Nublado	Alta	Normal	Falso	Sí
E14	lluvioso	Media	alta	Verdadero	No

- Regla de clasificación *SI Ambiente es nublado  
ENTONCES jugar = sí*
- Regla de asociación *SI Temperatura es media  
ENTONCES humedad es alta*



- Medidas utilizadas para evaluar una regla
  - Confianza y Soporte – Ayudan a determinar la relevancia y el interés de la regla

La **confianza** es la probabilidad condicional de que dado un evento A se produzca un evento B.

$$\text{Confianza } (A \rightarrow B) = P(B|A) = \frac{P(A \wedge B) P(B)}{P(A)}$$

Al hablar de reglas, la confianza se puede expresar como el porcentaje de ejemplos que satisfacen el antecedente y consecuente de la regla entre aquellos que satisfacen el antecedente.

El **soporte** se refiere al cociente del número de ejemplos que cumplen el antecedente y el consecuente de la regla entre el número total de ejemplos. En notación probabilística se puede expresar como:

$$\text{Soporte } (A \rightarrow B) = P(A \cup B)$$

Detectar aquellas reglas que, aunque se cumplen en algún caso y aunque puedan tener alta confianza, no son relevantes porque cubren casos poco frecuentes.

$$\text{Confianza (Regla 1 : ambiente = nublado} \rightarrow \text{jugar = si)} = \frac{4}{4} = 1$$

$$\text{Confianza (Regla 3 : temperatura = media} \rightarrow \text{humedad = alta)} = \frac{4}{6} = 0.67$$

$$\text{Soporte (Regla 1 : ambiente = nublado} \rightarrow \text{jugar = si)} = \frac{4}{14} = 0.29$$

$$\text{Soporte (Regla 3 : temperatura = media} \rightarrow \text{humedad = alta)} = \frac{4}{14} = 0.29$$

En un espacio con grandes cantidades de ejemplos, la combinación de reglas puede ser inmensa, por lo tanto, se establecen valores mínimos de confianza y soporte para considerar cuáles de las reglas aprendidas son relevantes.

- Lift

$$lift(A,B) = \frac{P(B \vee A)}{P(B)} = \frac{confianza(A \rightarrow B)}{P(B)}$$

El *lift* mide la correlación entre la ocurrencia de un hecho A y un hecho B:

- ▶ Si *lift* = 1, entonces el hecho A es independiente del hecho B.
- ▶ Si *lift* > 1, entonces existe correlación entre A y B y, por lo tanto, A probablemente implica B. Nos indica que la regla es útil.
- ▶ Si *lift* < 1, entonces existe correlación negativa entre A y B y, por lo tanto, A y B se comportan de forma opuesta (A probablemente implica no B). Nos indica que la regla no es útil.

- El aprendizaje de reglas de asociación comprenderá dos fases:

- 1 Encontrar aquellas reglas cuya frecuencia sea superior a un valor de soporte establecido.
- 2 De las reglas extraídas en el paso 1, seleccionar aquellas cuya confianza es superior a un valor determinado.

# Algoritmo de aprendizaje de reglas de asociación

## Algoritmo apriori

- El algoritmo apriori pretende generar ítem-sets que cumplan una cobertura mínima de manera eficiente. Un ítem es un par atributo-valor mientras que un ítem-set es un conjunto de pares atributo-valor.

El algoritmo apriori utiliza dos fases:

**FASE 1:** Generación de ítem-sets

**FASE 2:** Generación de reglas a partir de los ítems-sets generados en la fase 1

<i>Ítem-sets</i> de 1 elemento	Cobertura
Ambiente=soleado	5
Ambiente=nublado	4
Ambiente=lluvioso	5
Temperatura=alta	4

<i>Ítem-sets</i> de 2 elementos	Cobertura
Ambiente=lluvioso, temperatura=media	3
Ambiente=lluvioso, humedad=normal	3
Ambiente=soleado, humedad=alta	3
Ambiente=lluvioso, viento=falso	3
Ambiente=soleado, viento=falso	3

Ítem-sets de 3 elementos	Cobertura
Ambiente=soleado, humedad=alta, jugar=no	3
Ambiente=lluvioso, viento=falso, jugar=si	3
Temperatura=baja, humedad=normal, jugar=si	3
Humedad=normal, viento=falso, jugar=si	4

- Una vez obtenidos los *ítem-sets*, se procede a la siguiente fase del algoritmo que consiste en generar las reglas de asociación a partir de los *ítem-sets* encontrados de 3 y 2 elementos (para este ejemplo). De las reglas generadas, se descartan aquellas reglas que no superan un mínimo valor de confianza.

- Otros algoritmos
  - PART
  - *FP-Growth* y *TD-FP-Growth*
  - ECLAT
- Usos



# USOS

01

**Asociaciones entre artículos de la cesta de la compra**

02

**Análisis de sentimiento en texto**

03

**Diagnóstico médico a través de síntomas**

04

**Conocer los hábitos de usuarios de un gran centro deportivo**

05

**Asociaciones entre lugares visitados por turistas**

# Algoritmo a priori en Python

```
PS C:\Users\xxx> pip install mlxtend
```



Fuente: <http://rasbt.github.io/mlxtend/>

**MLxtend** (*Machine Learning extensions*) incluye extensiones útiles para realizar técnicas de *machine learning*, incluyendo, por ejemplo, el algoritmo *apriori*. Además,

- Otra librería que contiene el algoritmo apriori

```
from efficient_apriori import apriori
```

# Referencias

- <https://www.youtube.com/watch?v=IdvqFfMP7bl&t=199s>
- <https://www.youtube.com/watch?v=YRhu6yEseh8>

Gracias por su atención



[www.unir.net](http://www.unir.net)