

UNIVERSIDAD NACIONAL DE LA MATANZA

INTELIGENCIA DE NEGOCIOS

**Tecnologías Inteligentes
para Explotación de Información**

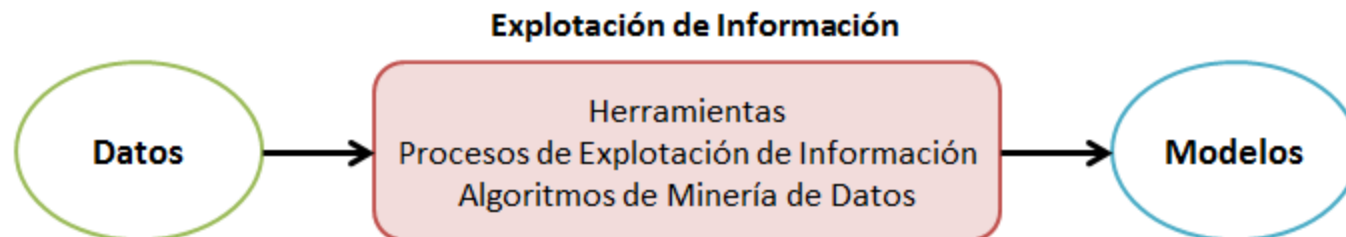
Profesor: Mg. Diego Basso

Curso 2017



USO DE LAS TECNOLOGÍAS

- ¿Cómo se usan las **tecnologías** para resolver un **problema**?
 - Tecnologías \Rightarrow Explotación de Información
 - Problema \Rightarrow Inteligencia de Negocio
- Construcción de modelos para descubrir conocimiento y soporte a la toma de decisiones:
 - Predictivos o Descriptivos
 - Entrenamiento + Prueba
 - Evaluación del modelo construido



TECNOLOGÍAS DE EXPLOTACIÓN DE INFORMACIÓN



- Basadas en Análisis Estadístico
 - Análisis de varianza
 - Regresión
 - Prueba Chi-cuadrado
 - Análisis de agrupamiento
 - Análisis de determinantes
 - Series de tiempo
- Basadas en Sistemas Inteligentes
 - Algoritmos de inducción TDIDT
 - Redes Neuronales SOM
 - Redes Bayesianas
 - Redes Neuronales Back-Propagation





MODELOS PREDICTIVOS

- Son modelos de aprendizaje supervisado que permiten predecir el resultado de variables de interés a partir de los valores de otras variables.
 - Variables de entrada \Rightarrow Atributos predictores
 - Variable a predecir \Rightarrow Atributo clase
- Se tiene un conjunto de *casos de entrenamiento* donde cada caso contiene un conjunto de **atributos** y uno de ellos es la **clase** a clasificar.
- Se separa un conjunto de *casos de prueba* para predecir nuevos casos y probar el modelo.
- Los nuevos casos deben ser asignados a su clase con la máxima exactitud y precisión posible.





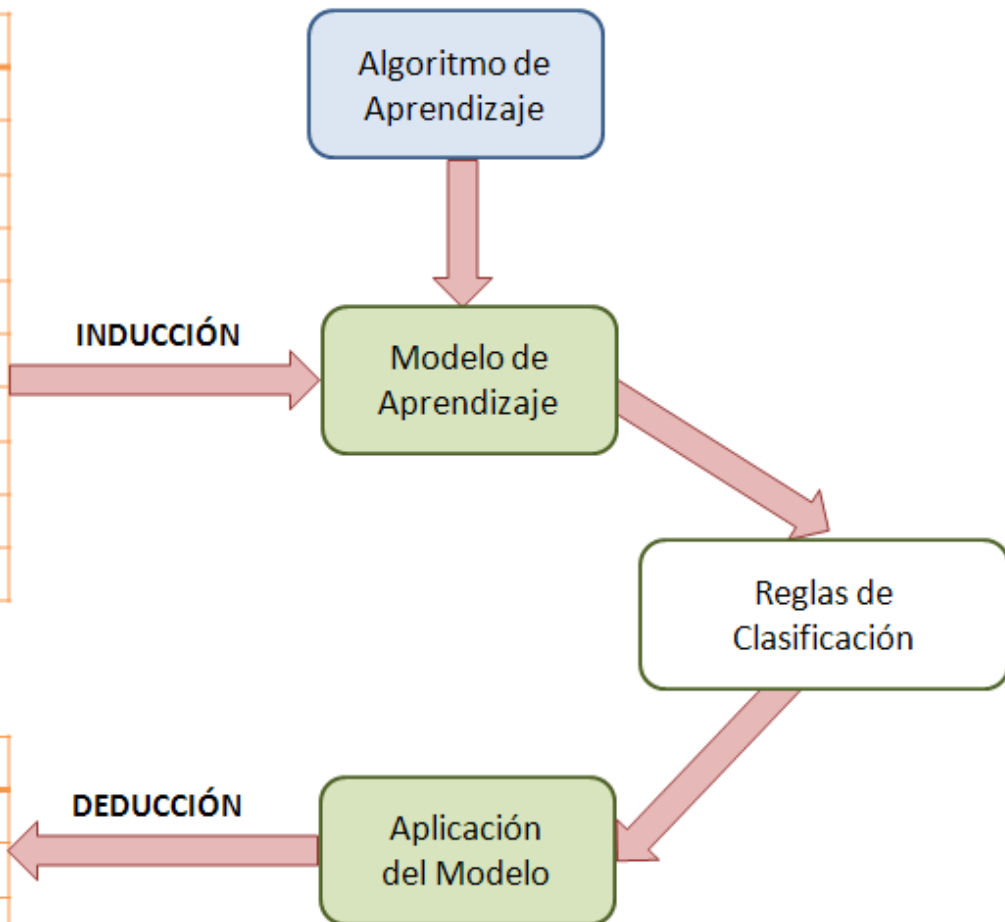
TAREA DE CLASIFICACIÓN

ATRIB 1	ATRIB 2	ATRIB 3	CLASE
Si	Large	125K	No
No	Medium	100K	No
No	Small	70K	No
Si	Medium	120K	No
No	Large	95K	Si
No	Medium	60K	No
Si	Large	220K	No
No	Small	85K	Si
No	Medium	75K	No
No	Small	90K	Si

Casos de Entrenamiento

ATRIB 1	ATRIB 2	ATRIB 3	CLASE
No	Small	55K	?
Si	Medium	80K	?
Si	Large	110K	?
No	Small	95K	?
No	Large	67K	?

Casos de Prueba





TECNOLOGÍAS PARA CLASIFICACIÓN

- Árboles de Decisión
 - Algoritmos de inducción TDIDT
 - Métodos basados en reglas
- Redes Bayesianas
 - Naïve-Bayes (Bayes Ingenuo)
- Vecinos más cercanos
 - K-Means





ALGORITMOS DE INDUCCIÓN TDIDT

- La familia TDIDT (*Top Down Induction Trees*) pertenece a los métodos inductivos del Aprendizaje Automático que aprenden a partir de ejemplos preclasificados.
 - Atributos **predictores** \Rightarrow Se particionan en diferentes ramas de acuerdo a los valores que el atributo puede tomar.
 - Pueden ser discretos o continuos.
 - Atributo **clase** \Rightarrow Decide la clase asignada (variable objetivo)
 - Debe ser discretizado.
- Generan árboles y reglas de decisión a partir de ejemplos preclasificados.





ALGORITMOS DE INDUCCIÓN TDIDT

- Se trata de identificar y ubicar en la parte superior del árbol a los atributos que mejor separan los ejemplos o muestras.
- Para encontrar los mejores atributos utiliza la teoría de la información, determinando qué atributo aporta la mayor ganancia de información (o menor pérdida de información) al tomar un determinado valor.
- Algoritmos utilizados ID3, C4.5 y C5

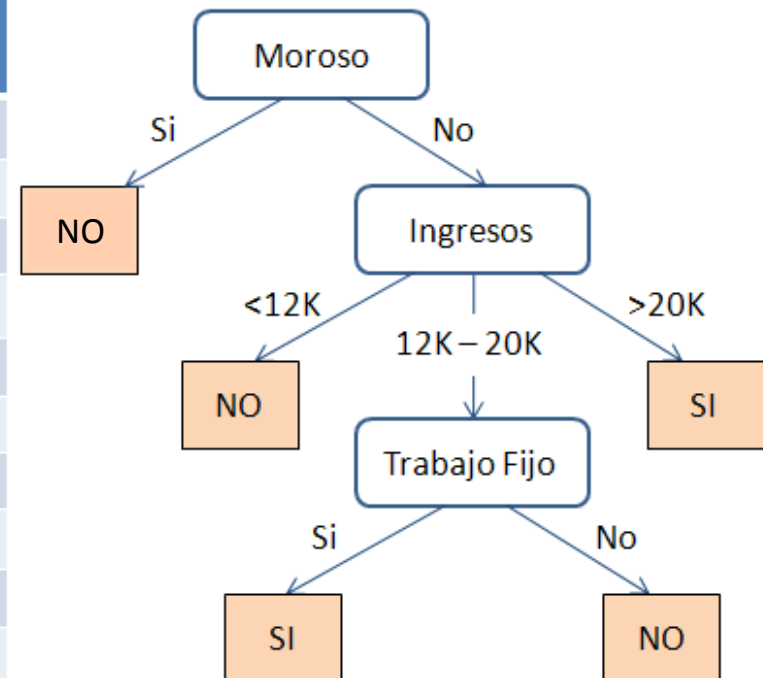




ALGORITMO TDIDT – EJEMPLO 1

- Presentación intuitiva del proceso de inducción.
- Evaluación de otorgamiento de préstamos a clientes
 - Atributo clase: **Otorgar Préstamo**

Ciente	Moroso	Antigüedad	Ingresos	Trabajo Fijo	Otorgar Préstamo
1	Si	> 5	12K – 20K	Si	No
2	No	< 1	12K – 20K	Si	Si
3	Si	1 - 5	> 20K	Si	No
4	No	> 5	> 20K	No	Si
5	No	< 1	> 20K	Si	Si
6	Si	1 - 5	12K – 20K	Si	No
7	No	1 - 5	> 20K	Si	Si
8	No	< 1	< 12K	Si	No
9	No	> 5	12K – 20K	No	No
10	Si	1 - 5	< 12K	No	No



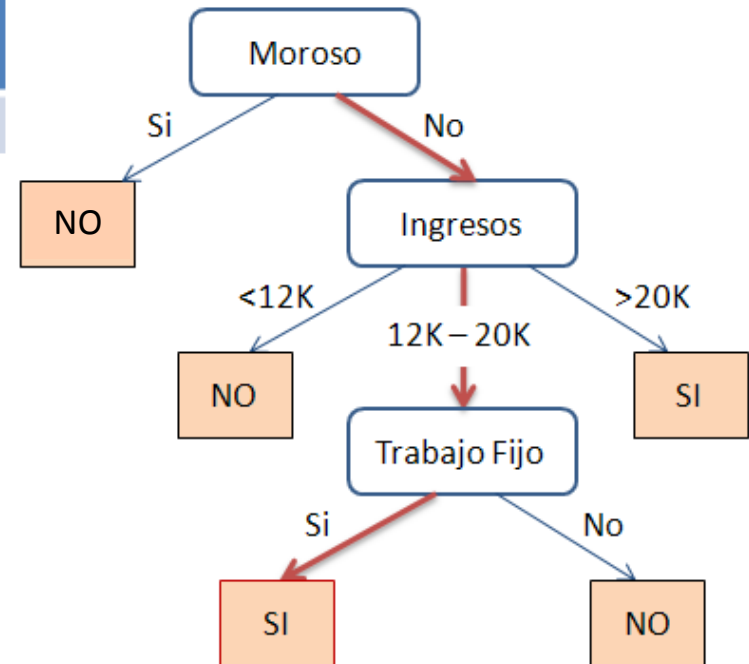
Casos de Entrenamiento



ALGORITMO TDIDT - APLICACIÓN

○ Caso de Prueba

Cliente	Moroso	Antigüedad	Ingresos	Trabajo Fijo	Otorgar Préstamo
11	No	1 - 5	12K – 20K	Si	?

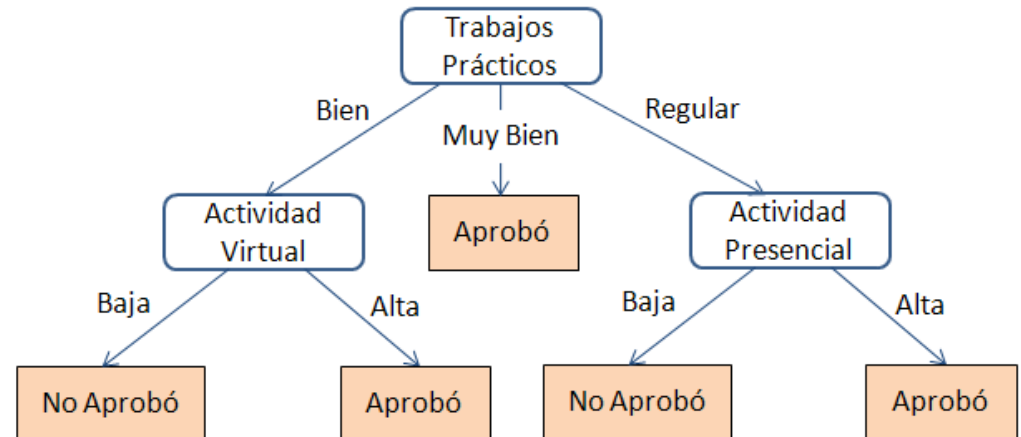




ALGORITMO TDIDT – EJEMPLO 2

- Predicción de resultados de exámenes
 - Atributo clase: **Resultado Parcial**

Trabajos Prácticos	Actividad Virtual	Actividad Presencial	Resultado Parcial
Bien	Alta	Alta	Aprobó
Bien	Baja	Alta	No Aprobó
Muy Bien	Alta	Alta	Aprobó
Regular	Alta	Alta	Aprobó
Regular	Alta	Baja	No Aprobó
Regular	Baja	Baja	No Aprobó
Muy Bien	Baja	Baja	Aprobó
Bien	Baja	Baja	No Aprobó
Bien	Alta	Baja	Aprobó
Regular	Baja	Baja	No Aprobó
Bien	Alta	Baja	Aprobó
Muy Bien	Alta	Alta	Aprobó
Regular	Baja	Baja	Aprobó
Regular	Alta	Alta	Aprobó





ALGORITMO TDIDT - APRENDIZAJE

○ Construcción de reglas del tipo **IF-THEN**

R₁: IF Trabajos Prácticos = 'Muy Bien' THEN Resultado Parcial = 'Aprobó'

R₂: IF (Trabajos Prácticos = 'Bien') AND (Actividad Virtual = 'Baja')
THEN Resultado Parcial = 'No Aprobó'

R₃: IF (Trabajos Prácticos = 'Bien') AND (Actividad Virtual = 'Alta')
THEN Resultado Parcial = 'Aprobó'

R₄: IF (Trabajos Prácticos = 'Regular') AND (Actividad Presencial = 'Baja')
THEN Resultado Parcial = 'No Aprobó'

R₅: IF (Trabajos Prácticos = 'Regular') AND (Actividad Presencial = 'Alta')
THEN Resultado Parcial = 'Aprobó'



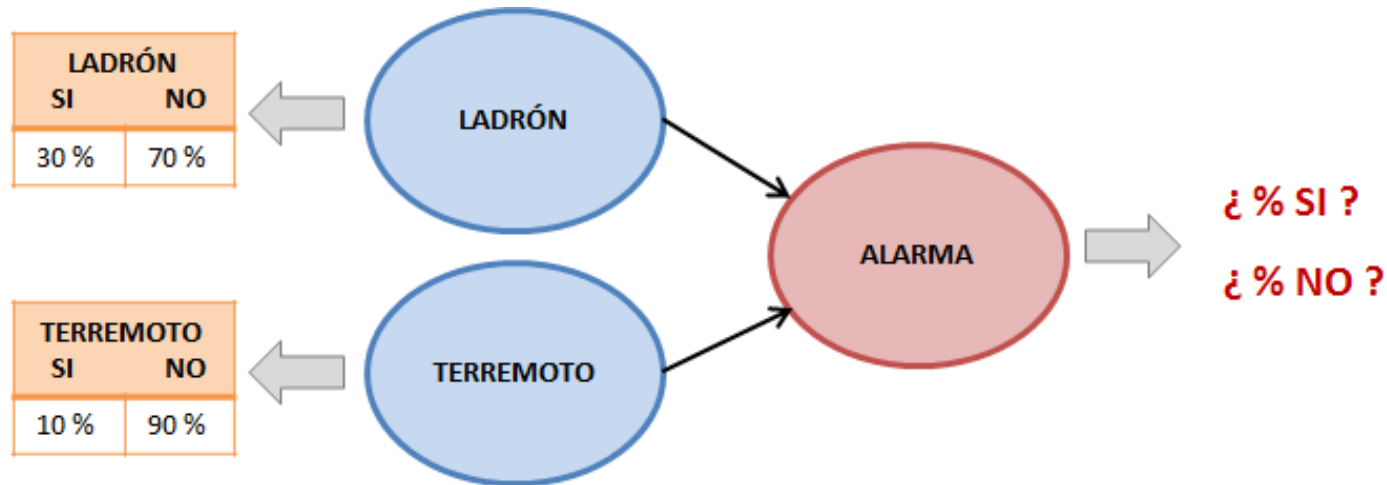


REDES BAYESIANAS

- Una red bayesiana es un grafo acíclico dirigido compuesto de nodos y arcos.
- Los nodos representan las variables aleatorias (o atributos).
- Los arcos representan dependencias probabilísticas de cada variable.
 - El arco entre dos variables, significa una influencia directa de una variable sobre otra.
 - Probabilidad condicional (Teorema de Bayes).
- Representan la relación causa-efecto entre atributos.
- Dan a una medida cuantitativa y probabilística de la importancia de los atributos en un problema de clasificación de clases.



REDES BAYESIANAS – EJEMPLO

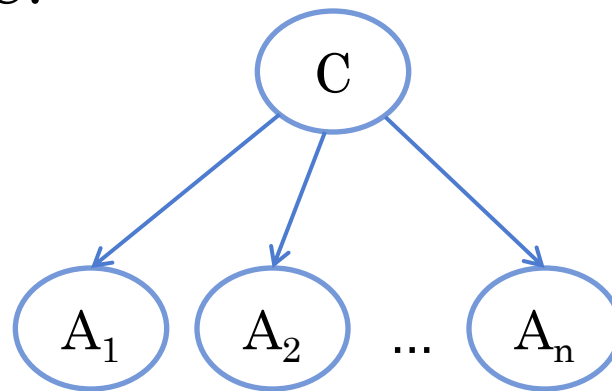


- Las variables *Ladrón* y *Terremoto* son causas para que se dispare una Alarma.
 - Existe una probabilidad a priori para Ladrón y Terremoto.
 - ¿Cuál es la probabilidad de que suene o no la alarma?
- *Ladrón* y *Terremoto* son condicionalmente independientes entre sí dada la variable *Alarma*.



CLASIFICADOR BAYESIANO NAÏVE-BAYES

- Considera que cada atributo predictor A_i y el atributo clase C son variables aleatorias.
- Las relaciones de dependencia entre los atributos A_i son condicionalmente independientes entre sí dado el atributo clase C .



- Dado un registro con atributos A_1, A_2, \dots, A_n el objetivo es predecir la clase C .
- Se busca encontrar el valor de C que maximice la probabilidad $p(C/A_1, A_2, \dots, A_n)$.



REDES BAYESIANAS

- Obtener una red bayesiana a partir de datos, es un proceso de aprendizaje.
 - Aprendizaje Estructural
 - Aprendizaje Paramétrico
- Proceso de inferencia
 - Predicciones a partir de observaciones





REDES BAYESIANAS – EJEMPLO

- Se tienen los siguientes datos:

	Ambiente	Temperatura	Humedad	Viento	Juega Tenis
1	Soleado	Alta	Alta	Leve	No
2	Soleado	Alta	Alta	Fuerte	No
3	Nublado	Alta	Alta	Leve	Si
4	Lluvioso	Media	Alta	Leve	No
5	Lluvioso	Baja	Normal	Fuerte	No
6	Lluvioso	Baja	Normal	Fuerte	No
7	Nublado	Baja	Normal	Leve	Si
8	Soleado	Media	Alta	Leve	Si
9	Soleado	Baja	Normal	Leve	Si
10	Lluvioso	Media	Normal	Leve	No
11	Soleado	Media	Normal	Fuerte	Si
12	Nublado	Media	Alta	Fuerte	Si
13	Nublado	Alta	Normal	Leve	Si
14	Lluvioso	Media	Alta	Fuerte	No





REDES BAYESIANAS – EJEMPLO

- Queremos saber si se jugará al tenis bajo las siguientes condiciones:

Ambiente	Temperatura	Humedad	Viento	Juega Tenis
Soleado	Baja	Alta	Fuerte	?

- El atributo clase a predecir es **Juega Tenis** cuyos valores serán **Si** o **No**.
- El nuevo caso será clasificado como clase C_j si $P(C_j) \prod_{i=1}^n P(A_i | C_j)$ es máximo.

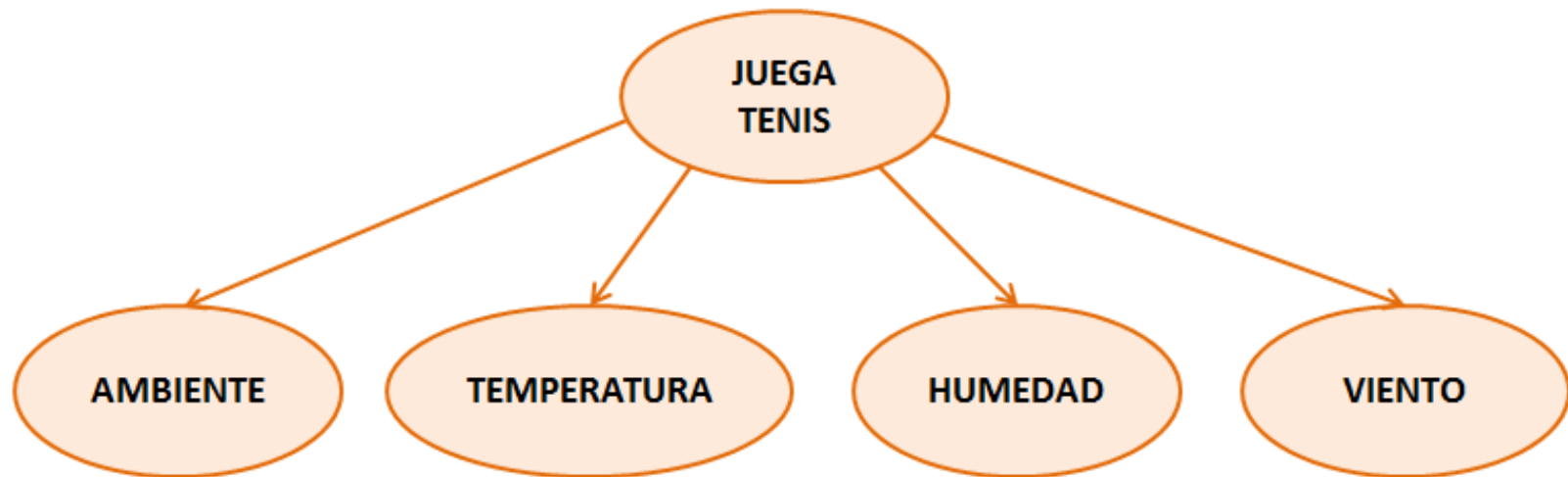




REDES BAYESIANAS – EJEMPLO

Aprendizaje Estructural

- Relaciones de dependencia e independencia





REDES BAYESIANAS – EJEMPLO

Aprendizaje Paramétrico

- Determinar probabilidades a priori de cada clase y las probabilidades condicionales.
- Analizando los 14 casos tenemos:

	Valores que toma	Cantidad de Casos	% casos totales
Ambiente	Soleado	5	35,7%
	Nublado	4	28,6%
	Lluvioso	5	35,7%
Temperatura	Alta	4	28,6%
	Media	6	42,8%
	Baja	4	28,6%
Humedad	Alta	7	50%
	Normal	7	50%
Viento	Leve	8	57,2%
	Fuerte	6	42,8%

Casos **Juega Tenis = Si** = 7

Casos **Juega Tenis = No** = 7

$P(\text{Juega Si}) = 0,5 = 50\%$

$P(\text{Juega No}) = 0,5 = 50\%$





REDES BAYESIANAS – EJEMPLO

- Desglosando los casos según si juegan o no al tenis:

Cantidad Casos			
	Valores que toma	Clase = Juega Tenis	
		Si	No
Ambiente	Soleado	3	2
	Nublado	4	0
	Lluvioso	0	5
Temperatura	Alta	2	2
	Media	3	3
	Baja	2	2
Humedad	Alta	3	4
	Normal	4	3
Viento	Leve	5	3
	Fuerte	2	4

Casos totales = 14

Casos **Juega Tenis = Si** = 7

Casos **Juega Tenis = No** = 7





REDES BAYESIANAS – EJEMPLO

- Obtenemos las probabilidades condicionales:

Probabilidades

	Valores que toma	Clase = Juega Tenis	
		Si	No
Ambiente	Soleado	$3/7 = 42,8\%$	$2/7 = 28,6\%$
	Nublado	$4/7 = 57,2\%$	0
	Lluvioso	0	$5/7 = 71,4\%$
Temperatura	Alta	$2/7 = 28,6\%$	$2/7 = 28,6\%$
	Media	$3/7 = 42,8\%$	$3/7 = 42,8\%$
	Baja	$2/7 = 28,6\%$	$2/7 = 28,6\%$
Humedad	Alta	$3/7 = 42,8\%$	$4/7 = 57,2\%$
	Normal	$4/7 = 57,2\%$	$3/7 = 42,8\%$
Viento	Leve	$5/7 = 71,4\%$	$3/7 = 42,8\%$
	Fuerte	$2/7 = 28,6\%$	$4/7 = 57,2\%$

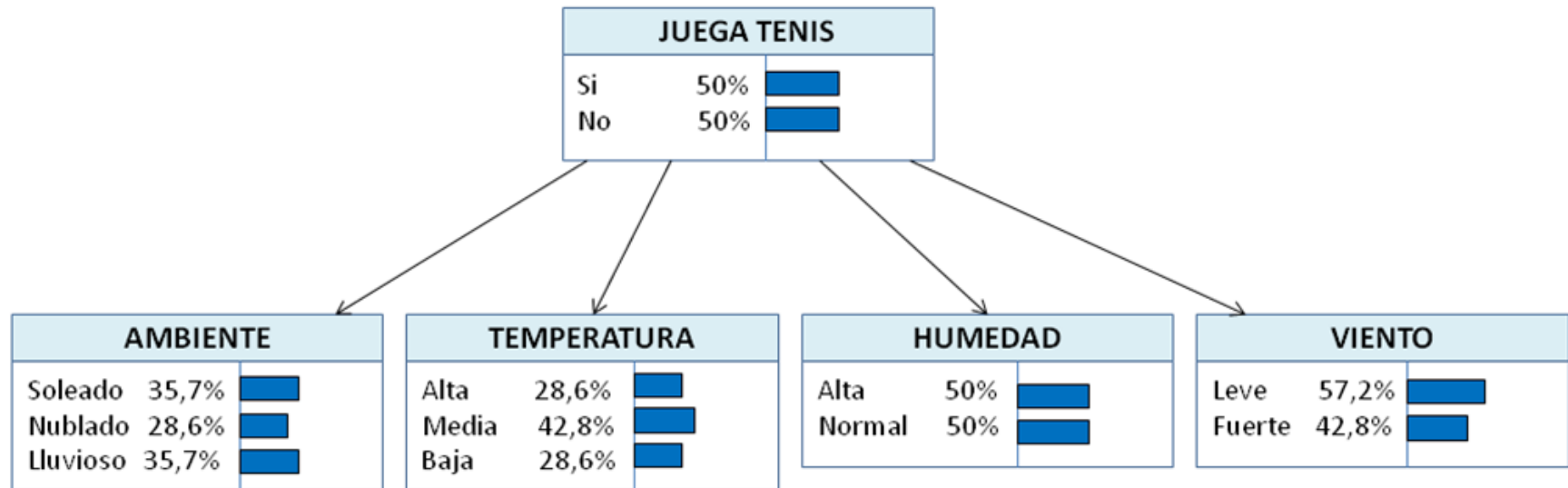
Casos totales = 14





REDES BAYESIANAS – EJEMPLO

Proceso de Inferencia

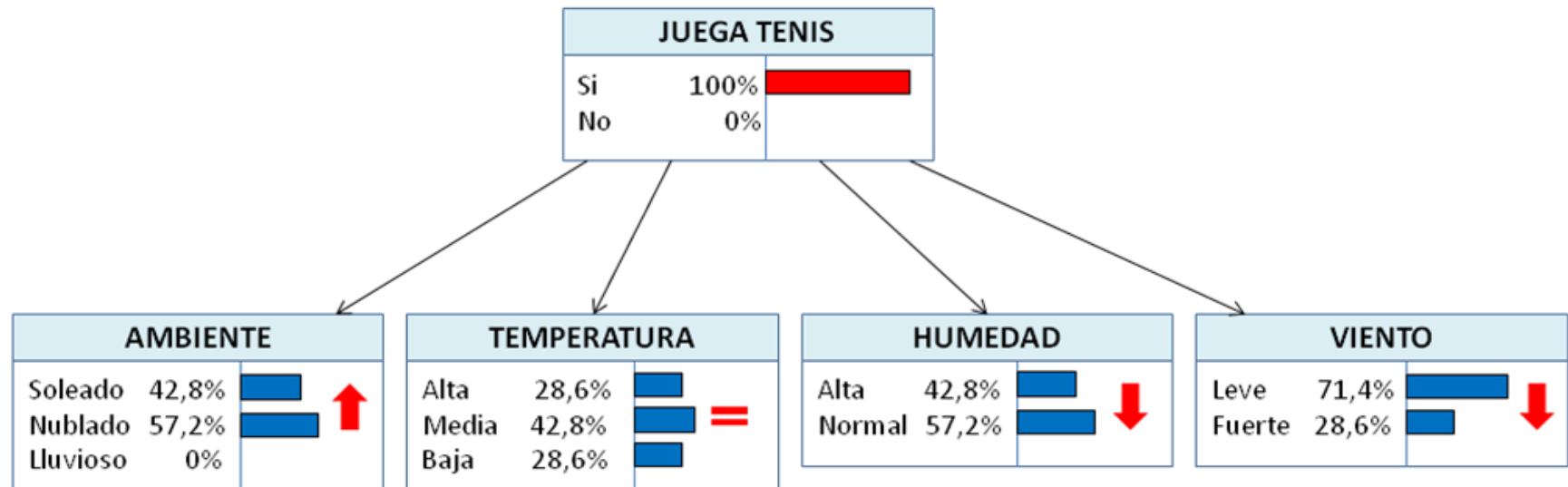




REDES BAYESIANAS – EJEMPLO

Proceso de Inferencia

- Juega Tenis = Si

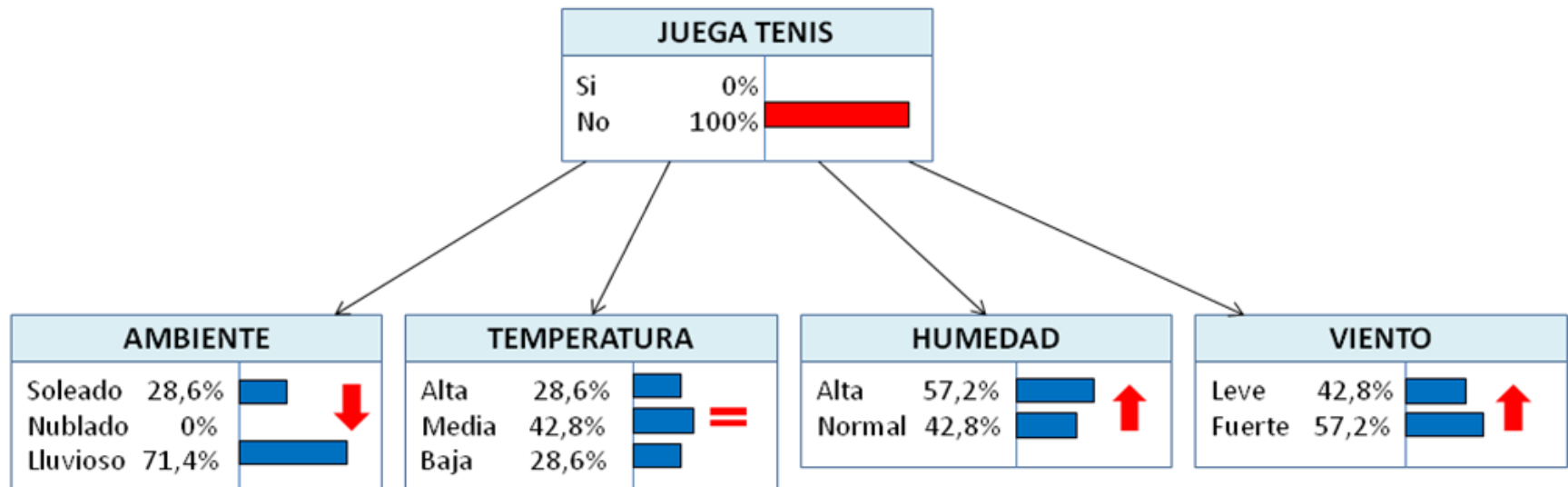




REDES BAYESIANAS – EJEMPLO

Proceso de Inferencia

- Juega Tenis = No





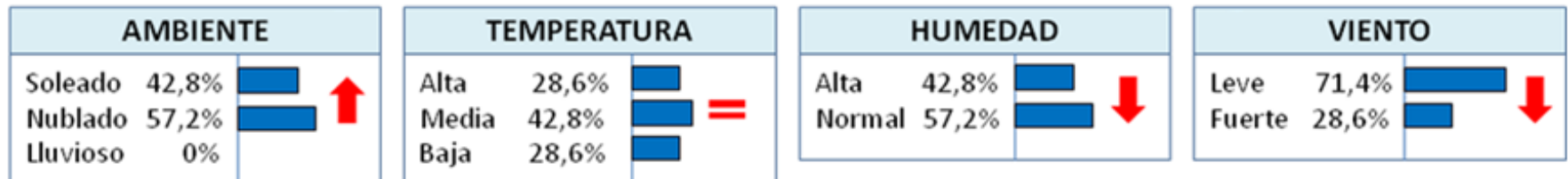
REDES BAYESIANAS – EJEMPLO

○ Predicción a realizar:

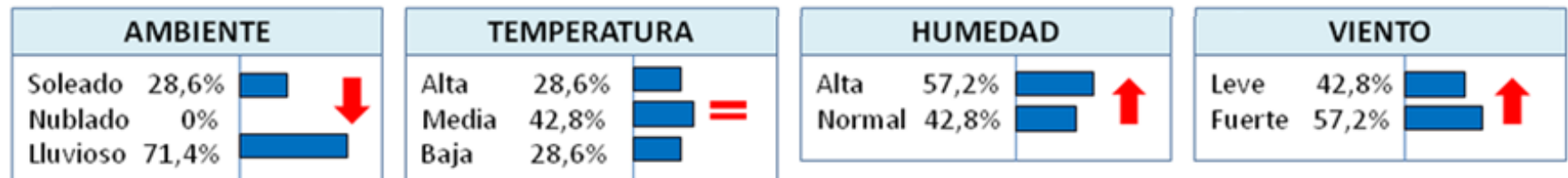
Ambiente	Temperatura	Humedad	Viento	Juega Tennis
Soleado	Baja	Alta	Fuerte	?

$$P(\text{Juega Si}) = 0,5$$
$$P(\text{Juega No}) = 0,5$$

- $P(\text{Juega Si}) = 0,428 \times 0,286 \times 0,428 \times 0,286 \times 0,5 = 0,0075$



- $P(\text{Juega No}) = 0,286 \times 0,286 \times 0,572 \times 0,572 \times 0,5 = 0,0133$





REDES BAYESIANAS – EJEMPLO

○ Normalizando:

- $P(\text{Juega Si}) = 0,0075 / (0,0075 + 0,0133) = 36\%$
- $P(\text{Juega No}) = 0,0133 / (0,0075 + 0,0133) = 64\%$

Ambiente	Temperatura	Humedad	Viento	Juega Tenis
Soleado	Baja	Alta	Fuerte	?

- El clasificador va a predecir que no se juega al tenis con una probabilidad del 64%.





MODELOS DE CLASIFICACIÓN

Métricas de Evaluación

- Se focalizan en analizar la capacidad de predicción y clasificación de clases del modelo construido.
- **Matriz de Confusión:** Permite comparar el resultado obtenido a partir del modelo predictivo construido con los resultados de los datos de prueba del modelo.
- Métricas utilizadas (en entrenamiento y prueba)
 - Exactitud del modelo
 - Precisión del modelo
- Otras métricas utilizadas
 - Cobertura de una regla
 - Precisión de una regla





MODELOS DE CLASIFICACIÓN

- Estructura de una **matriz de confusión** de 2 clases.

		Clase Clasificada		
		Clase A	Clase B	Total
Clase Real	Clase A	Nº casos clasificados como A y son de clase A (NCVA)	Nº casos clasificados como B pero son de clase A (NCFB)	Total de casos de la clase A
	Clase B	Nº casos clasificados como A pero son de clase B (NCFA)	Nº casos clasificados como B y son de clase B (NCVB)	Total de casos de la clase B
	Total	Total de casos clasificados como clase A	Total de casos clasificados como clase B	Número total de casos (NTC)

- Las métricas NCVA y NCVB representan los valores clasificados correctamente por el modelo.
- Las métricas NCFA y NCFB representan los errores (la confusión) entre las clases.





MODELOS DE CLASIFICACIÓN

- **Exactitud del modelo:** Proporción de casos clasificados correctamente respecto del número total de casos utilizados. Evalúa la capacidad de generalización del modelo para predecir y clasificar nuevos casos.

$$\text{Exactitud (M)} = \frac{\sum_{i=1}^n \text{NCV}_i}{\text{N}^\circ \text{ casos usados}}$$

- ALTA exactitud \Rightarrow Clasificaciones correctas $\geq 70\%$ casos.
- BAJA exactitud \Rightarrow Clasificaciones correctas $< 70\%$ casos. Modelo poco confiable.





MODELOS DE CLASIFICACIÓN

- **Precisión del modelo:** Proporción de casos reales de una clase respecto del total de casos clasificados por el modelo en esa misma clase. Evalúa la efectividad del modelo para clasificar casos a una clase particular.

$$\text{Precisión } (C_i) = \frac{NCV_i}{NCV_i + NCF_i}$$

- ALTA precisión \Rightarrow Modelo efectivo para predecir y clasificar nuevos casos.
- Precisión MEDIA \Rightarrow Modelo inestable. Posible confusión en clasificación y predicción.
- BAJA precisión \Rightarrow El modelo confunde las clases. Modelo poco efectivo.



MODELOS DE CLASIFICACIÓN - EJEMPLO



- Consideremos una matriz de confusión con 900 casos de clientes que pueden o no cerrar sus cuentas bancarias:
 - Clase SI – Se va del banco
 - Clase No – No se va del banco

		Clase Clasificada	
		Si	No
Clase Real	Si	455	29
	No	32	384

839 predicciones correctas

61 predicciones incorrectas

Exactitud (M) = 93,2%

Precisión (C_{No}) = 92,9%

Precisión (C_{Si}) = 93,4%



MODELOS DE CLASIFICACIÓN - EJEMPLO



- No todos los errores tienen el mismo costo para el banco.
 - El error en los falsos **No** (29 casos) es mucho más costoso para el banco ya que no se va a tomar ninguna acción y el cliente se va a ir.
 - El banco puede asignar un valor de costo a cada una de las celdas que representan un error en la clasificación para poder comparar soluciones de modelos.





MODELOS DESCRIPTIVOS

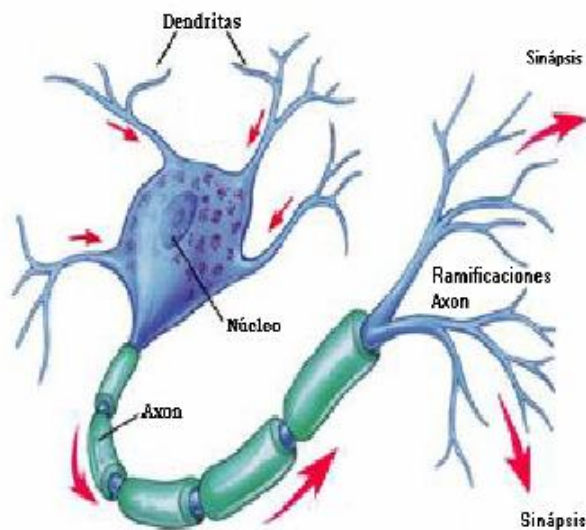
- No realizan predicciones.
- Analizan otros aspectos de los datos.
- Problemas tratados
 - Segmentación o Agrupamiento (Clustering)
 - Reglas de Asociación
- Tecnologías utilizadas
 - Redes Neuronales SOM
 - Algoritmo Apriori (descubrimiento de ítems frecuentes)



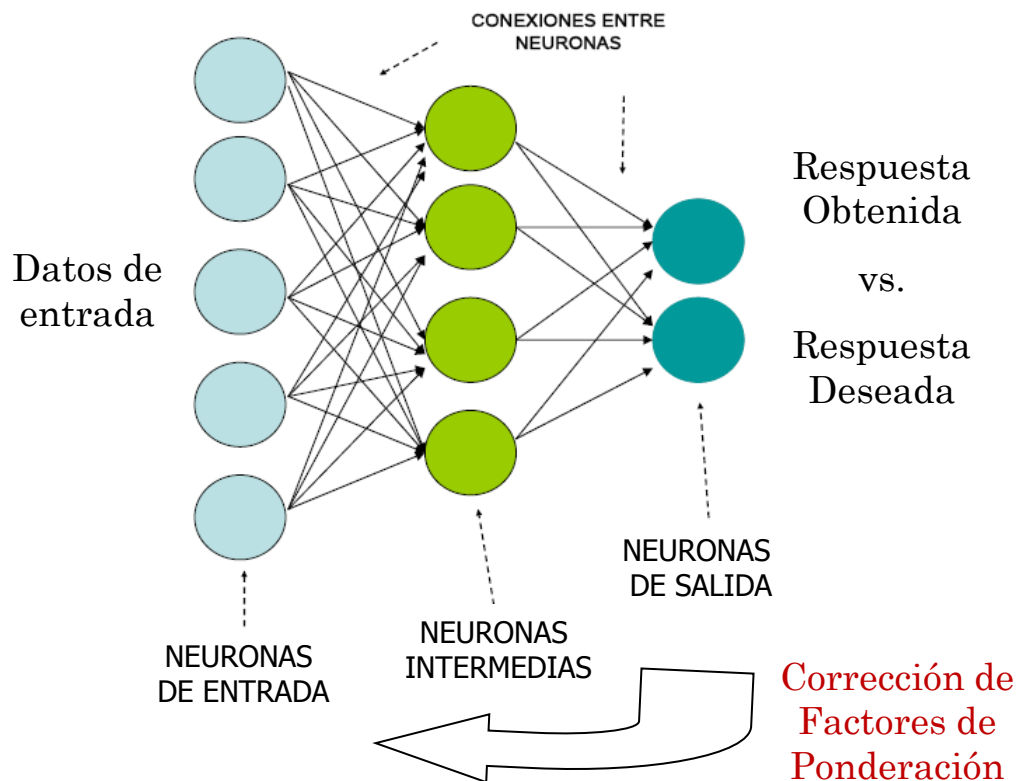


REDES NEURONALES

Neurona Biológica



Neurona Artificial



Casos de entrenamiento con sus respuestas
Casos de Prueba





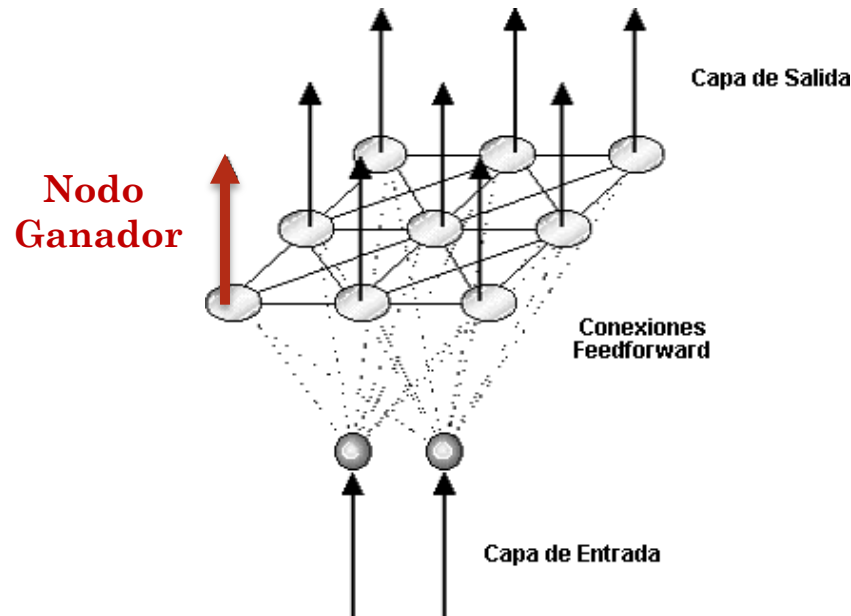
REDES NEURONALES SOM

- Mapas Auto-Organizados de Kohonen (SOM)
 - Entrenamiento no supervisado
 - Posee 2 capas de neuronas
 - Entrada – Un nodo por cada dato
 - Salida – Un nodo por cada categoría
- Produce una partición o segmentación (desconocida “apriori”) de los datos de entrada en categorías de datos con características similares (clusters).
- Los nodos de la capa de salida reciben impulsos de la capa de entrada.



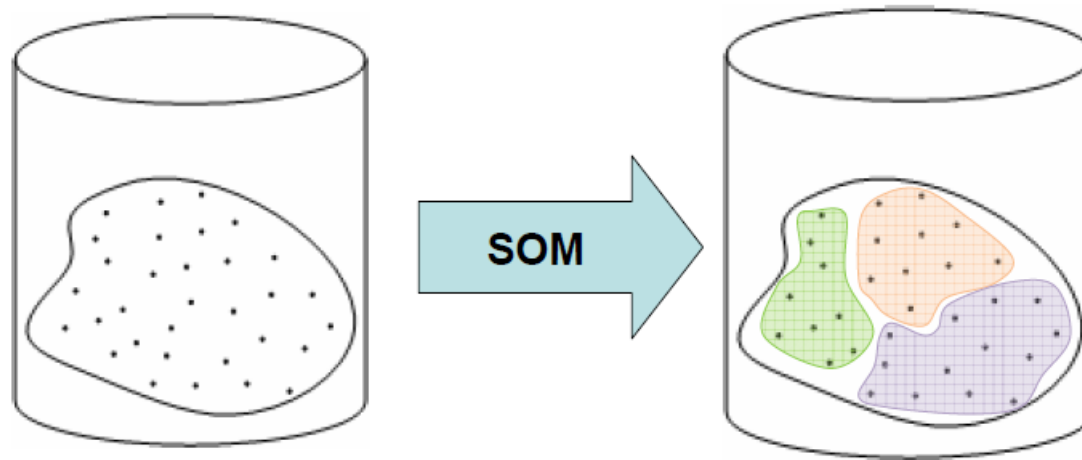


REDES NEURONALES SOM



- Cada vez que se presenta un registro de entrada, las neuronas “*compiten*” y una se define como la ganadora.
- Si se presenta un registro de entrada parecido al anterior, es muy posible que el ganador sea el mismo nodo de salida.

PROCESO INTUITIVO DE AGRUPAMIENTO



- Los registros semejantes van a parar a la misma categoría (cluster).
- Una vez entrenado, el mapa de Kohonen se puede usar para categorizar nuevos registros.





REGLAS DE ASOCIACIÓN

- Dado un conjunto de transacciones se quiere encontrar reglas que puedan predecir la ocurrencia de un ítem a partir de otros ítems de la transacción.
- Análisis de Canasta (**Market Basket Analysis**)

Ticket ID	Ítems
1	Pan, Leche
2	Pan, Pañales, Cerveza, Huevos
3	Leche, Pañales, Cerveza, Gaseosa
4	Pan, Leche, Pañales, Cerveza
5	Pan, Leche, Pañales, Gaseosa

Ejemplos de Reglas

{Pañales} \rightarrow {Cerveza}

{Cerveza, Pan} \rightarrow {Leche}

- Reglas de Asociación $X \rightarrow Y$ representan implicancias.
 - Antecedente \rightarrow Consecuente



REGLAS DE ASOCIACIÓN

Métricas de Evaluación

- **Cobertura de la Regla:** Proporción de casos a los que se le puede aplicar cada regla.

$$\text{COBER(R)} = \frac{\text{N}^\circ \text{casos que satisfacen la aplicación de la regla R}}{\text{N}^\circ \text{casos totales de la clase}}$$

- Toma valores entre 0 y 1.
- Reglas con mayor cobertura:
 - Representativas y útiles para obtener características que definen el comportamiento de una clase.
 - Credibilidad e interés del modelo para clasificar nuevos casos a una clase.
- Aplicable también a **modelos de clasificación.**





REGLAS DE ASOCIACIÓN - EJEMPLO

○ Ítems de compras:

Ticket ID	Ítems
1	Pan, Leche
2	Pan, Pañales, Cerveza, Huevos
3	Leche, Pañales, Cerveza, Gaseosa
4	Pan, Leche, Pañales, Cerveza
5	Pan, Leche, Pañales, Gaseosa

○ Consideremos las siguientes reglas:

- {Leche, Pañales} → {Cerveza} $\text{Cober(R)} = 2/5 = 0.4$
- {Leche, Cerveza} → {Pañales} $\text{Cober(R)} = 2/5 = 0.4$
- {Pañales, Cerveza} → {Leche} $\text{Cober(R)} = 2/5 = 0.4$
- {Cerveza} → {Leche, Pañales} $\text{Cober(R)} = 2/5 = 0.4$
- {Pan} → {Pañales, Gaseosa} $\text{Cober(R)} = 1/5 = 0.2$





REGLAS DE ASOCIACIÓN

- **Precisión de la Regla:** Proporción de casos que cumplen con la regla respecto del total de casos considerados en la precondition de la misma.

$$\text{Prec(R)} = \frac{\text{N}^\circ \text{casos que satisfacen la aplicación de la regla R}}{\text{N}^\circ \text{casos que satisfacen la precondition}}$$

- Toma valores entre 0 y 1.
- Cuanto mayor sea la precisión de una regla más confiable e interesante resulta para asociar ítems y descubrir nuevo conocimiento.
- Aplicable también a **modelos de clasificación**.





REGLAS DE ASOCIACIÓN – EJEMPLO

- Utilizando los mismos ítems de compra:

Ticket ID	Ítems
1	Pan, Leche
2	Pan, Pañales, Cerveza, Huevos
3	Leche, Pañales, Cerveza, Gaseosa
4	Pan, Leche, Pañales, Cerveza
5	Pan, Leche, Pañales, Gaseosa

- Considerando las reglas anteriores:

- {Leche, Pañales} → {Cerveza} $\text{Prec(R)} = 2/3 = 0.67$
- {Leche, Cerveza} → {Pañales} $\text{Prec(R)} = 2/2 = 1$
- {Pañales, Cerveza} → {Leche} $\text{Prec(R)} = 2/3 = 0.67$
- {Cerveza} → {Leche, Pañales} $\text{Prec(R)} = 2/3 = 0.67$
- {Pan} → {Pañales, Gaseosa} $\text{Prec(R)} = 1/4 = 0.25$





REGLAS DE ASOCIACIÓN

○ Conclusiones obtenidas:

- | | |
|--------------------------------|--|
| • {Leche, Pañales} → {Cerveza} | Cober(R) = 0.4 Prec(R) = 0.67 |
| • {Leche, Cerveza} → {Pañales} | Cober(R) = 0.4 Prec(R) = 1 |
| • {Pañales, Cerveza} → {Leche} | Cober(R) = 0.4 Prec(R) = 0.67 |
| • {Cerveza} → {Leche, Pañales} | Cober(R) = 0.4 Prec(R) = 0.67 |
| • {Pan} → {Pañales, Gaseosa} | Cober(R) = 0.2 Prec(R) = 0.25 |

○ La regla {Leche, Cerveza} → {Pañales} es la que mejor describe características de consumo, y la que genera mayor confianza e interés para descubrir patrones de compra frecuente.

○ Reglas con alta precisión pero baja cobertura son irrelevantes y de poco interés en un modelo de explotación de información.



PRÁCTICA CON TANAGRA

PREDICCIÓN / CLASIFICACIÓN



CASO PRÁCTICO – OFTALMOLOGÍA

- Archivo: Oftalmologia.xls
- Preguntas:
 - ¿Cuántos registros tiene el archivo?
 - ¿Cuántos atributos tiene y de qué tipo son?
 - ¿Cuál es la distribución de valores del atributo clase? ¿Está bien balanceado?
- Visualización de los datos

	EDAD	PADECIMIENTO	ASTIGMATISMO	LAGRIMEO	TIPO DE LENTE
1	joven	hipermétrope	si	reducido	ninguno
2	joven	hipermétrope	si	normal	duro
3	joven	hipermétrope	no	reducido	ninguno
4	joven	hipermétrope	no	normal	suave
5	joven	miope	si	reducido	ninguno
6	joven	miope	si	normal	duro
7	joven	miope	no	reducido	ninguno
8	joven	miope	no	normal	suave
9	pre-presbiópico	hipermétrope	si	reducido	ninguno
10	pre-presbiópico	hipermétrope	si	normal	ninguno
11	pre-presbiópico	hipermétrope	no	reducido	ninguno
12	pre-presbiópico	hipermétrope	no	normal	suave





CASO PRÁCTICO - OFTALMOLOGÍA

Algoritmos de Inducción TDIDT

○ Algoritmo a utilizar ID3

- Ejecutar el algoritmo con la parametrización por defecto.
- Evaluar el resultado obtenido.
- Visualizar el árbol de decisión.

○ Ajuste de parámetros

- Ejecutar el algoritmo con la nueva parametrización.
- Evaluar el resultado obtenido.
- Visualizar el árbol de decisión.

The screenshot shows a window titled "ID3 parameters" with a "Parameters" tab. It contains a "Tree stopping rule" section with four settings:

Parameter	Value
Min size for split :	5
Min size of leaves :	3
Max depth of the tree :	5
Min entropy gain for splitting :	0,0300

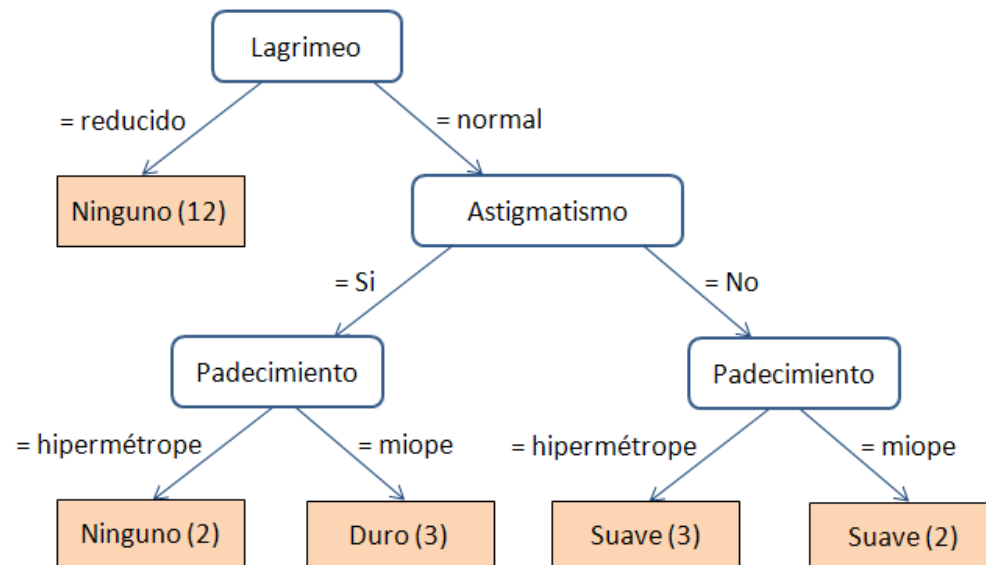


CASO PRÁCTICO - OFTALMOLOGÍA

○ Algoritmo ID3

Decision tree

- LAGRIMEO in [reducido] then TIPO DE LENTE = **ninguno** (100,00 % of 12 examples)
- LAGRIMEO in [normal]
 - ASTIGMATISMO in [si]
 - PADECIMIENTO in [hipermétrope] then TIPO DE LENTE = **ninguno** (66,67 % of 3 examples)
 - PADECIMIENTO in [miope] then TIPO DE LENTE = **duro** (100,00 % of 3 examples)
 - ASTIGMATISMO in [no]
 - PADECIMIENTO in [hipermétrope] then TIPO DE LENTE = **suave** (100,00 % of 3 examples)
 - PADECIMIENTO in [miope] then TIPO DE LENTE = **suave** (66,67 % of 3 examples)





CASO PRÁCTICO - OFTALMOLOGÍA

○ Evaluación del algoritmo ID3

- N° casos entrenamiento: **24 casos**
 - Ninguno = 15 casos / Duro = 4 casos / Suave = 5 casos

	ninguno	duro	suave	Sum
ninguno	14	0	1	15
duro	1	3	0	4
suave	0	0	5	5
Sum	15	3	6	24

Exactitud (M) = 91,7%

Precisión (C_{Ninguno}) = 93,3%

Precisión (C_{Duro}) = 100%

Precisión (C_{Suave}) = 83,3%

- N° reglas obtenidas: **5**
- Cobertura (R_i) > 100 / N° reglas $\text{Cober}(R_i) > 20\%$

Clase (Ninguno)	Clase (Duro)	Clase (Suave)
$\text{Cober}(R_1) = 12/15 = 80\%$	$\text{Cober}(R_3) = 3/4 = 75\%$	$\text{Cober}(R_4) = 3/5 = 60\%$
$\text{Cober}(R_2) = 2/15 = 13,3\%$		$\text{Cober}(R_5) = 2/5 = 40\%$





CASO PRÁCTICO - OFTALMOLOGÍA

○ Algoritmo a utilizar C4.5

- Ejecutar el algoritmo con la parametrización por defecto.
- Evaluar el resultado obtenido.
- Visualizar el árbol de decisión.

○ Ajuste de parámetros

- Ejecutar el algoritmo con la nueva parametrización.
- Evaluar el resultado obtenido.
- Visualizar el árbol de decisión.

The screenshot shows a window titled "C4.5 parameters" with a "Parameters" tab. It contains two settings:

- "Min size of leaves" is set to 3, with a spin button to its right.
- "Confidence level" is set to 0,25, with a text input field to its right.



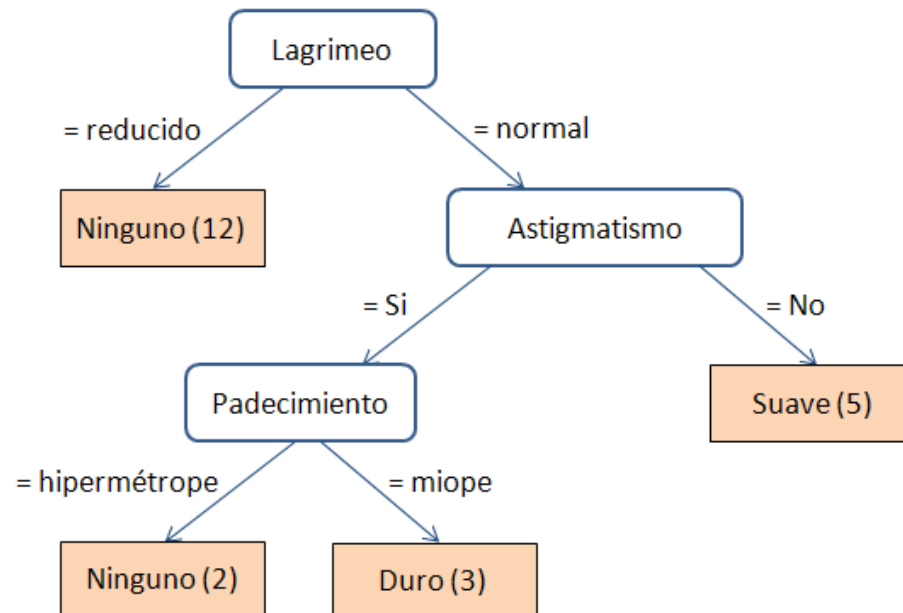


CASO PRÁCTICO - OFTALMOLOGÍA

○ Algoritmo C4.5

Decision tree

- LAGRIMEO in [reducido] then TIPO DE LENTE = **ninguno** (100,00 % of 12 examples)
- LAGRIMEO in [normal]
 - ASTIGMATISMO in [si]
 - PADECIMIENTO in [hipermétrope] then TIPO DE LENTE = **ninguno** (66,67 % of 3 examples)
 - PADECIMIENTO in [miope] then TIPO DE LENTE = **duro** (100,00 % of 3 examples)
 - ASTIGMATISMO in [no] then TIPO DE LENTE = **suave** (83,33 % of 6 examples)





CASO PRÁCTICO - OFTALMOLOGÍA

○ Evaluación del algoritmo C4.5

	ninguno	duro	suave	Sum
ninguno	14	0	1	15
duro	1	3	0	4
suave	0	0	5	5
Sum	15	3	6	24

Exactitud (M) = 91,7%

Precisión (C_{Ninguno}) = 93,3%

Precisión (C_{Duro}) = 100%

Precisión (C_{Suave}) = 83,3%

- N° reglas obtenidas: 4
- Cobertura (R_i) > 100 / N° reglas $\text{Cober}(R_i) > 25\%$

Clase (Ninguno)	Clase (Duro)	Clase (Suave)
$\text{Cober}(R_1) = 12/15 = 80\%$	$\text{Cober}(R_3) = 3/4 = 75\%$	$\text{Cober}(R_4) = 5/5 = 100\%$
$\text{Cober}(R_2) = 2/15 = 13,3\%$		

- El algoritmo C4.5 es el que mejor predice cómo clasificar.

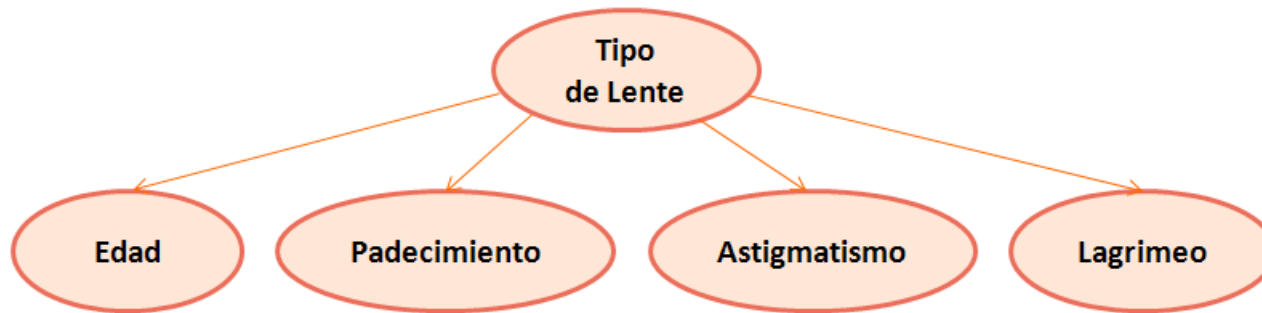




CASO PRÁCTICO - OFTALMOLOGÍA

Clasificación con Redes Bayesianas

○ Aprendizaje Estructural



○ Algoritmo a utilizar Naïve-Bayes

- Ejecutar el algoritmo con la parametrización por defecto.
- Evaluar el resultado obtenido.





CASO PRÁCTICO - OFTALMOLOGÍA

○ Evaluación del algoritmo

	ninguno	duro	suave	Sum
ninguno	14	0	1	15
duro	0	4	0	4
suave	0	0	5	5
Sum	14	4	6	24

Exactitud (M) = 95,8%

Precisión (C_{Ninguno}) = 100%

Precisión (C_{Duro}) = 100%

Precisión (C_{Suave}) = 83,3%

○ Aprendizaje Paramétrico

- Probabilidades a priori para cada Tipo de Lente

Prior distribution of class attribute "TIPO DE LENTE"

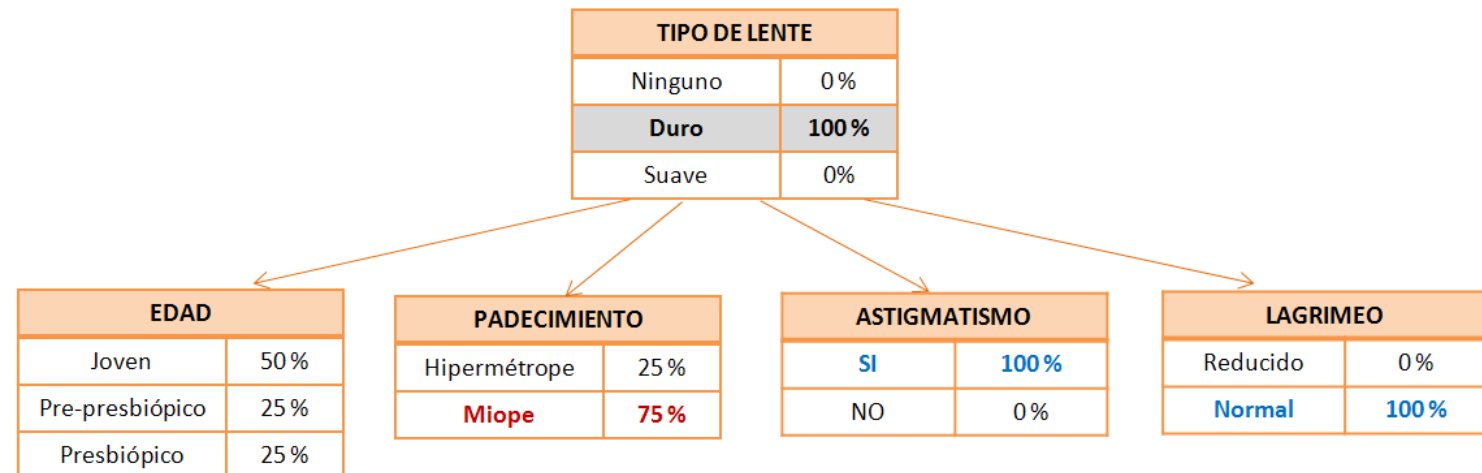
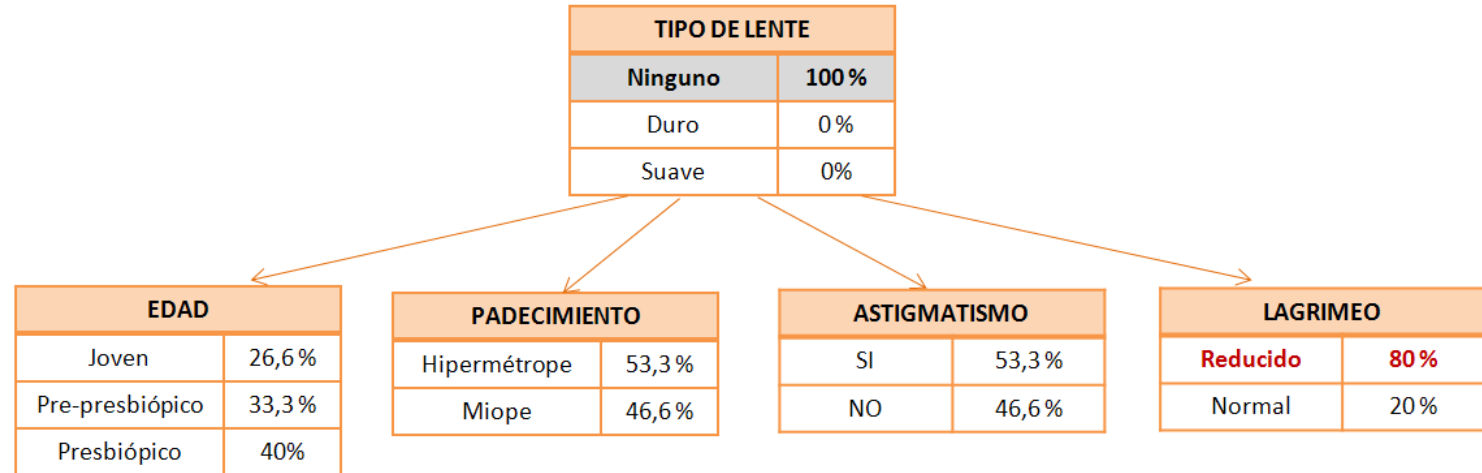
Values	Count	Percent	Histogram
ninguno	15	62,50 %	
duro	4	16,67 %	
suave	5	20,83 %	





CASO PRÁCTICO - OFTALMOLOGÍA

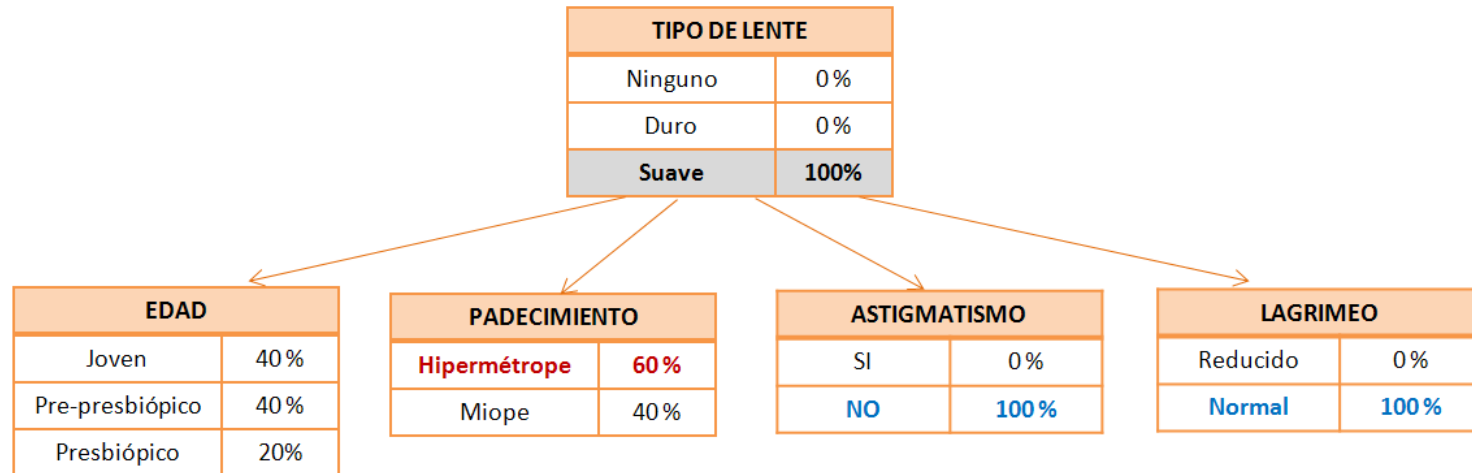
○ Proceso de Inferencia





CASO PRÁCTICO - OFTALMOLOGÍA

○ Proceso de Inferencia



○ ¿Qué conocimiento obtenemos en base a los resultados del análisis de cada tipo de lente?



