Como aprenden las máquinas

bienvenida

Universitat d'Alacant
Universidad de Alicante
UNIVERSITAT POLITÈCNICA DE VALÈNCIA
UJI UNIVERSITAT JAUME I
valgrAI

Sesión 13:
Sistemas PLN. Métodos estadísticos, basados en reglas (no ML, LLM).

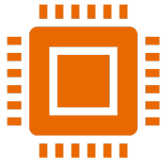
bienvenida

Indice

1. Introducción
2. Sistemas basados en reglas
3. Métodos estadísticos

Si lo que quieres saber...

¿Cómo hacemos que las máquinas entiendan el lenguaje humano?

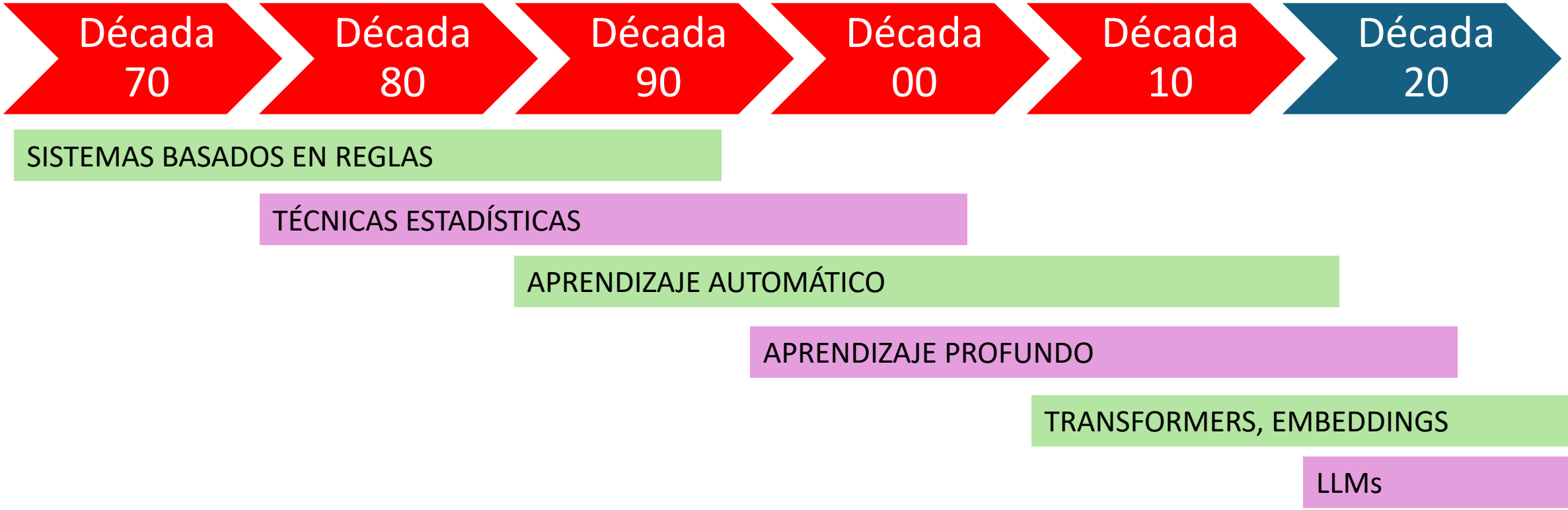


Un ordenador convencional basa su forma de ”**aprender**” en codificar y decodificar información digital binaria basada en ceros y unos. Para que una máquina “**entienda**” nuestro lenguaje, debemos de convertir el texto en códigos binarios. Esto se conoce como **Text Encoding.**

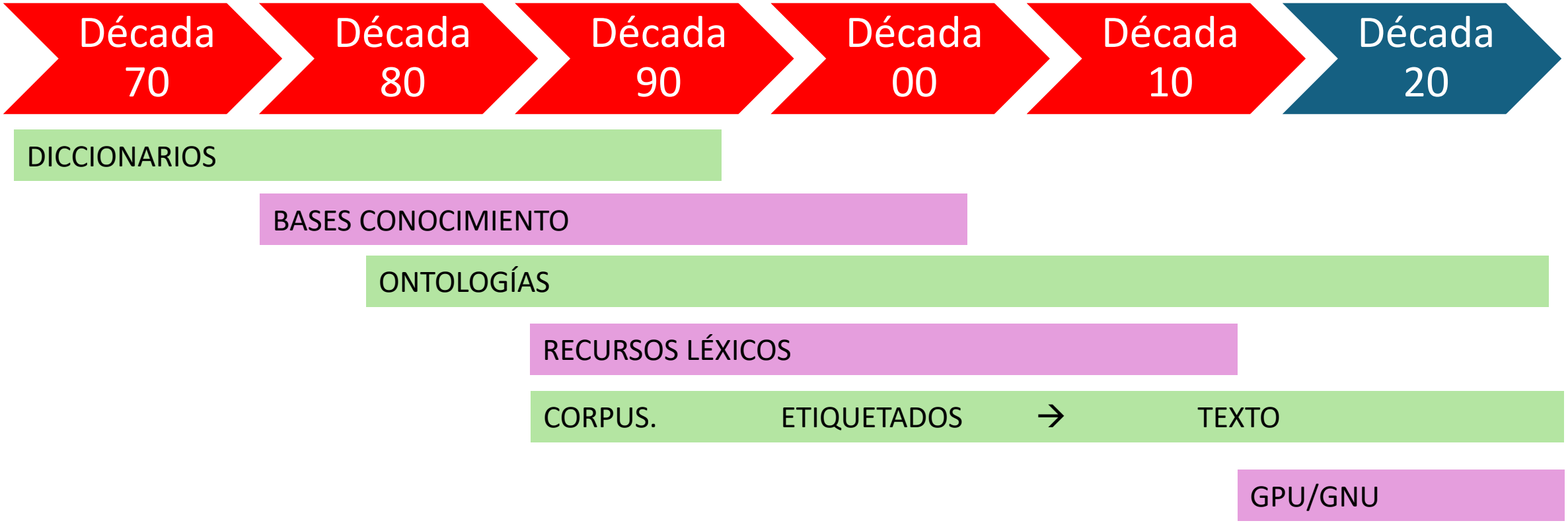
Métodos de convertir texto en códigos binarios:

1. Métodos sencillos – **Reglas**
2. Métodos complejos y modernos basados en IA – **Métodos estadísticos, aprendizaje automático, Deep learning, modelos de lenguaje**

Evolución NLP: Técnicas



Evolución NLP: Recursos



Evolución NLP: Aplicaciones

ANALIZADORES LÉXICOS Y SINTÁCTICOS

APLICACIONES INTERMEDIAS

TRADUCCION AUTOMÁTICA

RECUPERACIÓN Y EXTRACCIÓN DE INFORMACIÓN

CLASIFICACIÓN, PRIMERAS APLICACIONES INDUSTRIALES

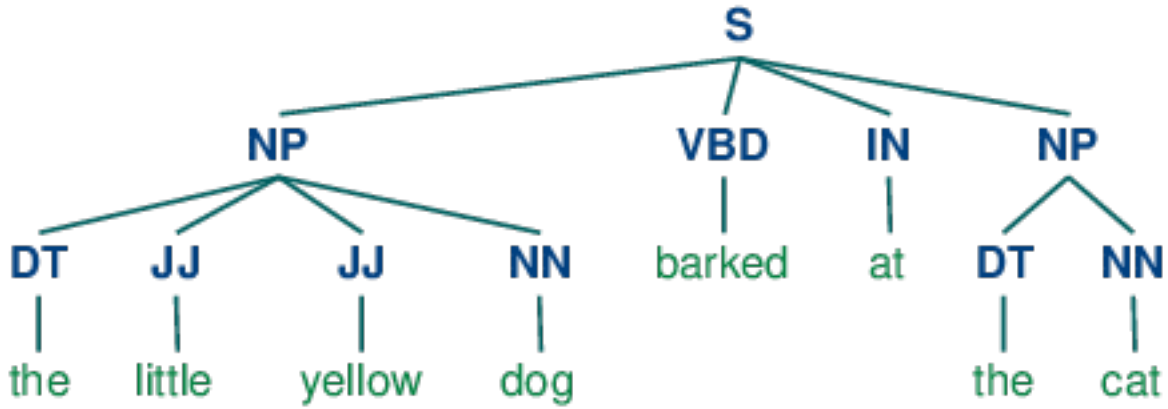
SISTEMAS COVERSACIONALES

HUMANIZAR IA

MÉTODOS BASADOS EN REGLAS

Métodos basados en reglas

- enfoque clásico del PLN que utiliza un conjunto de reglas predefinidas para analizar y procesar el lenguaje natural.
- Estas reglas se basan en la lingüística y el conocimiento experto para determinar la estructura y el significado del texto.



Métodos basados en reglas

Ventajas

- **Precisión:** muy precisos para tareas específicas, como la identificación de entidades nombradas o la clasificación de textos.
- **Explicabilidad:** Las reglas son fáciles de entender y explicar.
- **Eficiencia:** Los sistemas basados en reglas pueden ser muy eficientes, especialmente para tareas bien definidas.

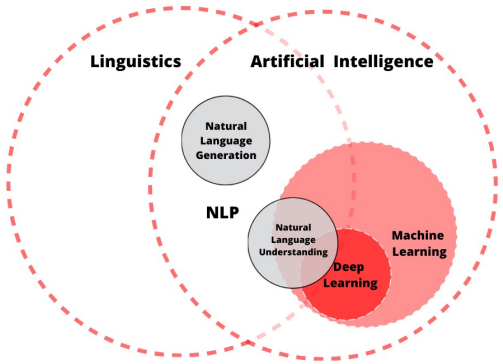
Inconvenientes

- **Falta de flexibilidad:** difíciles de adaptar a nuevos tipos de texto o dominios.
- **Dificultad de desarrollo:** La creación y el mantenimiento de un conjunto de reglas extenso puede ser un proceso largo y costoso.
- **Falta de generalización:** pueden tener dificultades para generalizar a nuevos casos que no están cubiertos por las reglas existentes

Métodos basados en reglas

Tareas

- **Analizadores sintácticos:** Los analizadores sintácticos utilizan reglas para determinar la estructura sintáctica de una oración.
- **Sistemas de clasificación de textos.** Estos sistemas funcionan mediante un conjunto de reglas predefinidas que se utilizan para determinar la categoría a la que pertenece un texto. Las reglas se basan en características lingüísticas del texto, como palabras clave, la estructura sintáctica y la semántica.
- **Sistemas de traducción automática:** Los sistemas de traducción automática basados en reglas utilizan reglas para traducir texto de un idioma a otro.
- **Sistemas de extracción de información:** Los sistemas de extracción de información utilizan reglas para identificar y extraer información específica de un texto.



Métodos basados en reglas

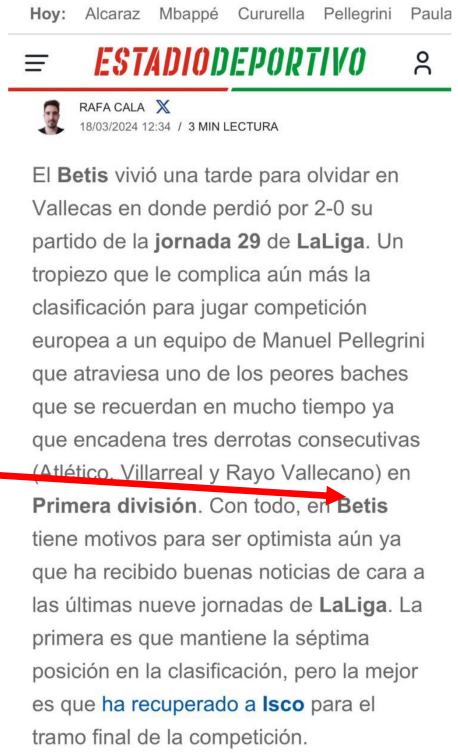
Tipos

- **Diccionarios, Gazzetters, BoW (Bag of words)**
- **Reglas de dominio, Patrones**
- **Sistemas híbridos**

Métodos basados en reglas

BoW (Bag of Words)

Equipos de futbol



Hoy: Alcaraz Mbappé Cururella Pellegrini Paula

ESTADIODEPORTIVO

RAFA CALA
18/03/2024 12:34 / 3 MIN LECTURA

El **Betis** vivió una tarde para olvidar en Vallecas en donde perdió por 2-0 su partido de la **jornada 29** de **LaLiga**. Un tropiezo que le complica aún más la clasificación para jugar competición europea a un equipo de Manuel Pellegrini que atraviesa uno de los peores baches que se recuerdan en mucho tiempo ya que encadena tres derrotas consecutivas (Atlético, Villarreal y Rayo Vallecano) en **Primera división**. Con todo, el **Betis** tiene motivos para ser optimista aún ya que ha recibido buenas noticias de cara a las últimas nueve jornadas de **LaLiga**. La primera es que mantiene la séptima posición en la clasificación, pero la mejor es que ha recuperado a **Isco** para el tramo final de la competición.

¿qué problemas piensas que tiene estos sistemas?

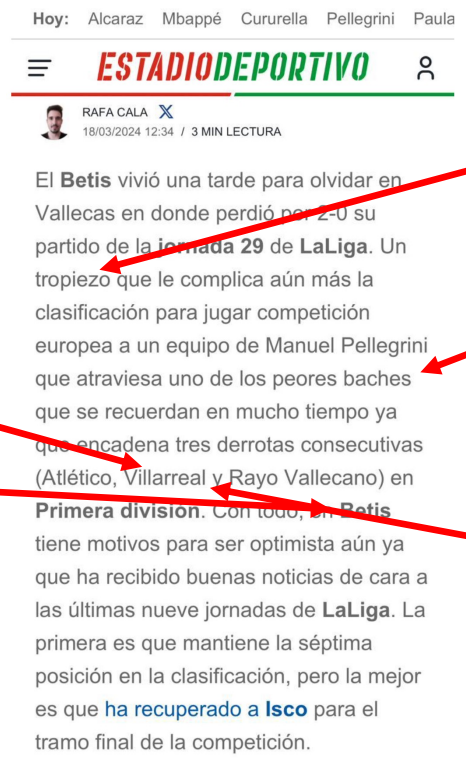
1. Pertenencia a varias categorías
2. Incompletitud

Métodos basados en reglas

BoW (Bag of Words): Problema pertenencia varias categorias

Equipos de futbol

Villareal, Barcelona, Atlético de Madrid, Betis, ... Hércules



Hoy: Alcaraz Mbappé Cururella Pellegrini Paula

ESTADIODEPORTIVO

RAFA CALA
18/03/2024 12:34 / 3 MIN LECTURA

El **Betis** vivió una tarde para olvidar en Vallecas en donde perdió por 2-0 su partido de la **jornada 29** de **LaLiga**. Un tropiezo que le complica aún más la clasificación para jugar competición europea a un equipo de Manuel Pellegrini que atraviesa uno de los peores baches que se recuerdan en mucho tiempo ya que encadena tres derrotas consecutivas (Atlético, Villarreal y Rayo Vallecano) en **Primera división**. Con todo, el **Betis** tiene motivos para ser optimista aún ya que ha recibido buenas noticias de cara a las últimas nueve jornadas de **LaLiga**. La primera es que mantiene la séptima posición en la clasificación, pero la mejor es que ha recuperado a **Isco** para el tramo final de la competición.

Sucesos

tropiezos, asesinatos, ... baches

ciudades

Villarreal, Madrid, ... Sevilla

Métodos basados en reglas

BoW

- Ventajas
  - Rápido/fácil de construir
- Inconvenientes
  - Polisemia
  - Sinonimia
  - No tiene en cuenta el contorno
  - Incompletitud

Métodos basados en reglas

REGLAS

- Expresiones regulares
- Heurísticas
- Reglas

Métodos basados en reglas

EXPRESIONES REGULARES

- Una expresión regular es un tipo de patrón que puede aplicarse a un texto.
- Si una expresión regular coincide con una parte del texto, entonces puede averiguar fácilmente qué parte
- Una expresión regular o bien coincide con el texto (o con una parte del texto), o bien no coincide.
- Las expresiones regulares son extremadamente útiles para procesar texto

Métodos basados en reglas

EXPRESIONES REGULARES

- La expresión regular **"[a-z]+"** coincidirá con una secuencia de una o más letras minúsculas
  - **[a-z]** significa cualquier carácter de la a a la z, ambas inclusive
  - **+** significa "uno o más".
- Supongamos que aplicamos este patrón a la cadena ”**Hoy tenemos clase**”
- Hay tres formas de aplicar este patrón:
  - **A toda la cadena**: no coincide porque la cadena contiene caracteres que no son minúsculas
  - **Al principio de la cadena**: no coincide porque la cadena no empieza por minúscula.
    - Para buscar en la cadena: tendrá éxito y coincidirá con “**oy**”
    - Si el patrón se aplica por segunda vez, encontrará “**tenemos**”
    - Posteriomente encontrará, “**clase**”
    - Después fallará

Métodos basados en reglas

EXPRESIONES REGULARES: PATRONES SIMPLES

- **abc** exactamente esta secuencia de tres letras
- **[abc]** cualquiera de las letras a, b, o c
- **[^abc]** cualquier carácter excepto una de las letras a, b o c (inmediatamente dentro de un corchete abierto, ^ significa "no", pero en cualquier otro lugar sólo significa el carácter ^)
- **[a-z]** cualquier carácter de la a a la z, ambas inclusive
- **[a-zA-Z0-9]** cualquier letra o dígito

Métodos basados en reglas

EXPRESIONES REGULARES: SECUENIAS Y ALTERNATIVAS

- Si un patrón va seguido de otro, los dos patrones deben coincidir consecutivamente
- Por ejemplo, [A-Za-z]+[0-9] coincidirá con una o más letras seguidas inmediatamente por un dígito.

GPLSI1 GpLsi2 GPlsi3 gplsi4

- La barra vertical "|" se utiliza para separar alternativas. El patrón **pL|SI** coincidirá con "**pL**" o "**SI**".

GPLSI1 GpLsi2 GPlsi3 gplsi4

Métodos basados en reglas

EXPRESIONES REGULARES: CARACTERES PREDEFINIDOS

`.`	Cualquier caracter menos el fin de línea
`\d`	un digito: `[0-9]`
`\D`	no es digito: `[^0-9]`
`\s`	espacio: `[ \t\n\x0B\f\r]`
`\S`	no es espacio: `[^\s]`
`\w`	palabra: `[a-zA-Z_0-9]`
`\W`	no es palabra: `[^\w]`

Métodos basados en reglas

EXPRESIONES REGULARES: LÍMITES

Estos patrones coinciden con la cadena vacía si se encuentra en la posición especificada:

^	el principio de una línea
$	el final de una línea
\b	un límite de palabra
\B	no es un límite de palabra
\A	el principio de la entrada (pueden ser varias líneas)
\Z	el final de la entrada excepto el terminador final, si hay
\z	el final de la entrada
\G	el final de la coincidencia anterior

Métodos basados en reglas

EXPRESIONES REGULARES: Cuantificadores

- Supongamos que **X** representa algún patron

X?	opcional, X aparece una o ninguna vez
X*	X aparece cero o más veces
X+	X aparece una o más veces
X{n}	X ocurre exactamente n veces
X{n,}	X ocurre n veces o más
X{n,m}	X ocurre al menos n veces pero no más de m veces

- Todos estos operadores son postfijos, es decir, vienen después del operando

Métodos basados en reglas

EXPRESIONES REGULARES: CUANTIFICADORES

Cuidado con los cuantificadores : supongamos el texto **aardvark**

- Usando el patrón **a*ardvark** (a* maximiza):
  - La **a*** coincidirá primero con **aa**, pero entonces **ardvark** no coincidirá.
  - A continuación, **a*** "retrocede" y sólo coincide con una **a**, permitiendo que el resto del patrón (**ardvark**) tenga éxito. **¿QUÉ EVITA?** **aaarvarl**
- Usando el patrón **a*?ardvark** (a*? es redundante):
  - La **a*?** coincidirá primero con cero caracteres (la cadena nula), pero entonces **ardvark** no coincidirá.
  - A continuación, **a*?** se extiende y coincide con la primera **a**, permitiendo que el resto del patrón (**ardvark**) tenga éxito.
- Utilizando el patrón **a*+ardvark** (a*+ es posesivo):
  - La **a*+** coincidirá con la **aa**, y no retrocederá, por lo que **ardvark** nunca coincidirá y el patrón fallará.

Métodos basados en reglas

EXPRESIONES REGULARES: GRUPOS

En las expresiones regulares, los **paréntesis** se utilizan para **agrupar**, pero también capturan (guardan para su uso posterior) todo lo que coincida con esa parte del patrón

Ejemplo: **([a-zA-Z]*)([0-9]*)** coincide con cualquier número de letras seguido de cualquier número de dígitos

Si la coincidencia es correcta, **\1 ([a-zA-Z]*)** contiene las letras coincidentes y

\2 ([0-9]*) los dígitos coincidentes.

Además, **\0 ([a-zA-Z]*)([0-9]*)** contiene todo lo que coincide con el patrón completo.

Los grupos de captura se numeran contando sus paréntesis de apertura de izquierda a derecha: **((A) (B (C)))**

1 2 3 4

\0 = \1 = ((A)(B(C))), \2 = (A), \3 = (B(C)), \4 = (C)

Métodos basados en reglas

EXPRESIONES REGULARES: DOBLE BARRA INVERTIDA

- Las barras invertidas tienen un significado especial en las expresiones regulares; por ejemplo, **\b** significa el límite de una palabra.
- Si se desea encontrar una cadena con una barra invertida debe usarse ****
  - Si escribe **”([a-z])+([0-9)+\b", no** obtendrá una cadena con caracteres de barra invertida, que no es lo que desea. **expediente34\b**
  - Pero si escribe **”([a-z])+([0-9)+\\b", no** obtendrá una cadena con caracteres de barra invertida, que no es lo que desea. **expediente34\b**

Supongamos que desea buscar la secuencia de caracteres “a*” (**una a seguida de una estrella**)

"a*" y **"a*"**; no funciona; significa "cero o más as”.

"a*" sí funciona; es la cadena de tres caracteres **a, \, *.**

Métodos basados en reglas

EXPRESIONES REGULARES: EJERCICIO

- Realiza la expression regular que reconozca en el texto una dirección de correos
  - Rafael@gmail.com ([A-Za-z0-9]+)@([A-Za-z0-9]+).([A-Za-z0-9]+)
  - **rafael@dlsi.ua.es** ([A-Zaa-Z0-9]+)@([A-Zaa-Z0-9]+).([A-Zaa-Z0-9]+).]+).([A-Zaa-Z0-9]+)
  - **Rafael.munoz@ua.es**
  - **Rafael_munoz@ua.es**

- **([A-Za-z._]+)@([A-Za-z.]+)**

Métodos basados en reglas

REGLAS: GRAMÁTICAS

- Las reglas son desarrolladas de forma manual por un experto del dominio de aplicación.
- Si se desea construir un analizador sintáctico se usará las reglas gramaticales del idioma sobre el que se aplicará el analizador

• S = verbo	Corre!
• S = SN + verbo	Yo paseo
• S = SN + verbo + SP	El perro mordió al ladrón
• SN = Pronombre	
• SN = articulo + nombre	
• SP = …….	
• …….	

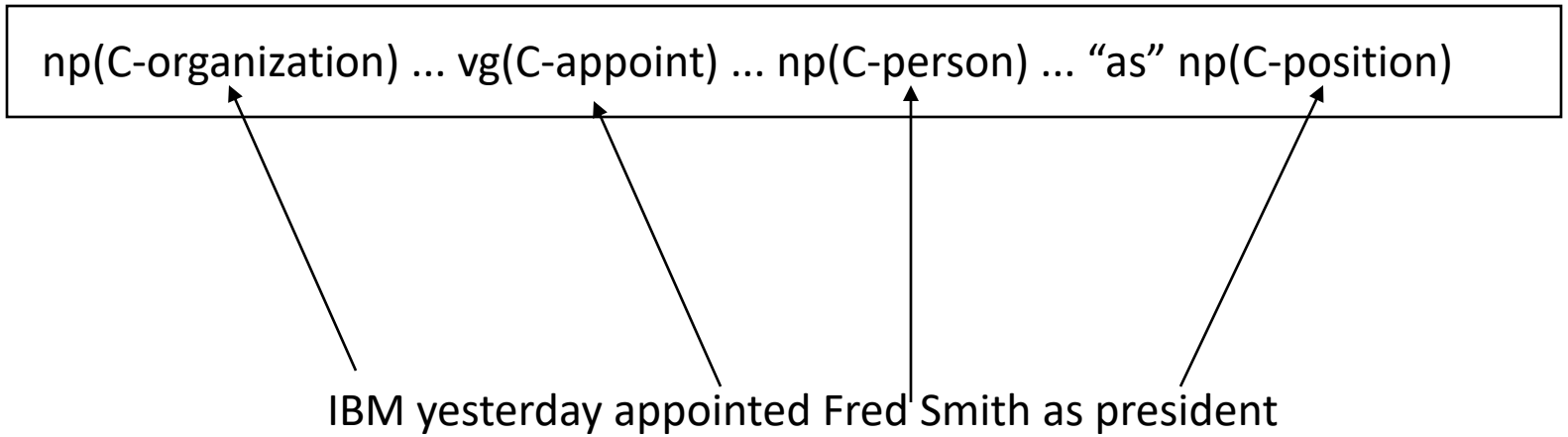
Métodos basados en reglas

REGLAS: GRAMÁTICAS

- Tipos de reglas, 3 niveles:
  - bajo nivel: gran aplicabilidad (normalmente incluidos en el sistema)
  - intermedio: librerías de patrones (aplicables a diferentes dominios)
    - ej. extractores de entidades (persona, empresa, lugar, organización)
    - extractores de relaciones (persona/oranización, organización/lugar)
  - específicos del dominio

Métodos basados en reglas

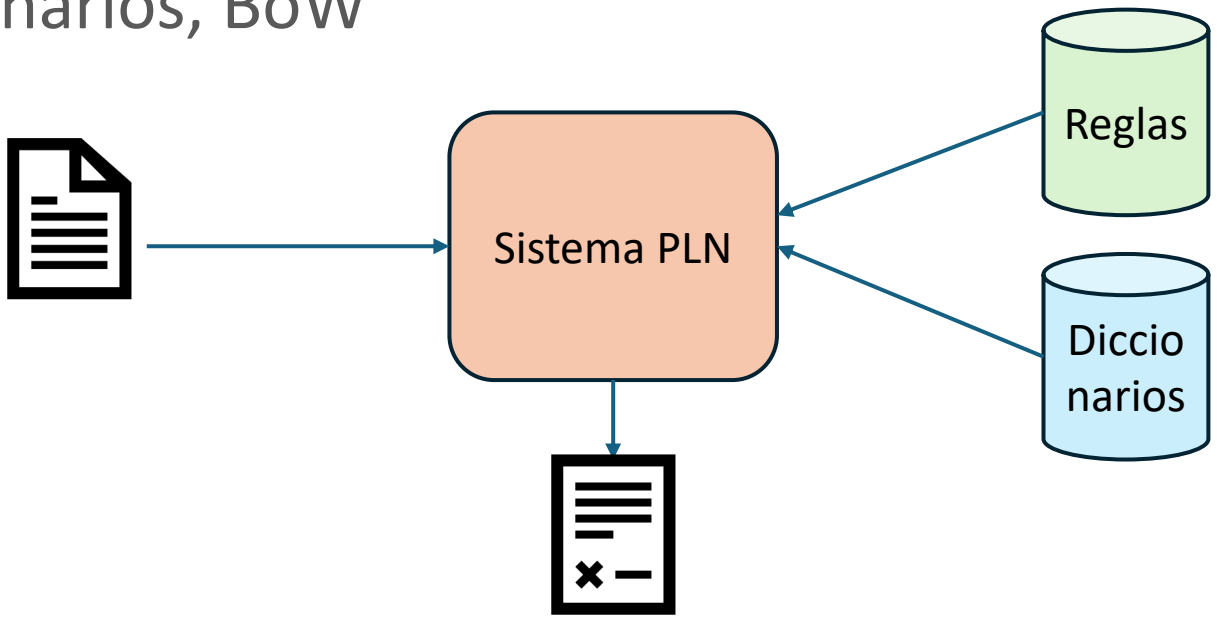
Patrones



Métodos basados en reglas

SISTEMAS HÍBRIDOS

- Utilización de reglas junto con recursos léxicos: Ontologías, Diccionarios, BoW



Métodos basados en reglas

Ejercicio

- https://regex101.com
- https://corenlp.run
- CLIPS

- Weka https://waikato.github.io/weka-wiki/downloading_weka/

Métodos basados en reglas

Ejercicio

- https://regex101.com

- Reconocer nombre de personas

Métodos estadísticos

Tipos de problemas

Clasificación

Regresión lineal

Agrupación

Paradigmas de Aprendizaje

Supervisados

No supervisados

Refuerzo

Métodos estadísticos

Tipos de problemas: CLASIFICACIÓN

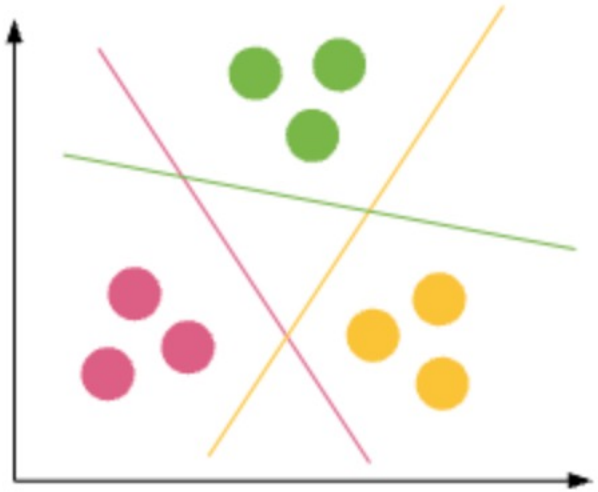
Encontrar f dado un conjunto de tuplas (V, C) donde V son variables o características y C es una clase

$$f(X)=C$$

Algoritmos más usados: regresión logística, SVC, KNN C

Ejemplos:

- Correos spam o no spam
- Clientes buenos o malos



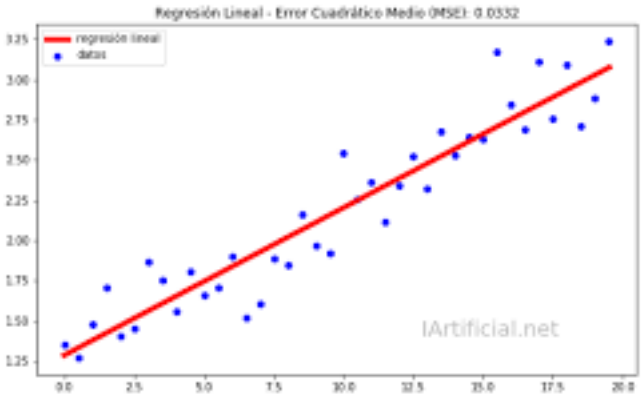
Métodos estadísticos

Tipos de problemas: REGRESIÓN LINEAL

Encontrar f dado un conjunto de tuplas (V, C) donde V son variables o características y C es un número

$f(X)=C$

Algoritmos más usados: regresión lineal, KNN C



Ejemplos:

- Precio de una vivienda V(superficie, barrio, …) C= valor
- Salario de una persona V(nivel de estudio, sector,…) C=Valor

Métodos estadísticos

Tipos de problemas: AGRUPAMIENTO

Encontrar f dado un conjunto V = variables obtengan particiones de V

Es como una clasificación, pero sin clases predefinidas

Algoritmos más usados: K-Means, clustering jerarquico

Ejemplos:

- Segmentar clientes para campaña marketing
- Segmentar documentos

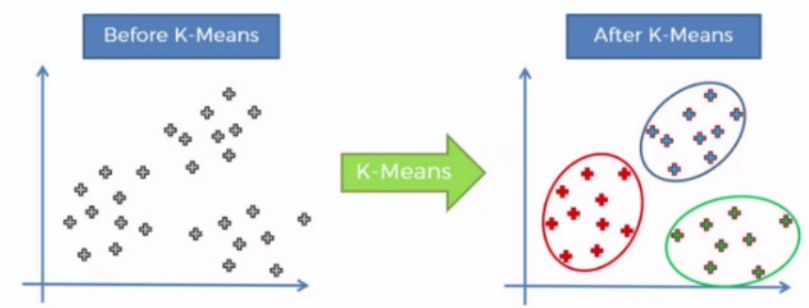


Imagen vía Towards Data Science.

Métodos estadísticos

Clasificación completa de modelos estadísticos en PLN

I. Modelos Generativos

II. Modelos Discriminativos

III. Modelos para Tareas de Clasificación

IV. Modelos para Tareas de Regresión

V. Modelos para Tareas de Traducción

VI. Otros Modelos

Métodos estadísticos

Clasificación completa de modelos estadísticos en PLN

I. Modelos Generativos

- **Modelos de Soporte Vectorial:**
  - **Ventajas:** Buenos para tareas de clasificación binaria.
  - **Desventajas:** No son tan buenos para tareas de clasificación multiclase.
- **Redes Neuronales Convolucionales:**
  - **Ventajas:** Buenos para tareas de extracción de entidades nombradas.
  - **Desventajas:** No son tan buenos para tareas de modelado del lenguaje.
- **Redes Neuronales Recurrentes:**
  - **Ventajas:** Buenos para tareas de modelado del lenguaje.
  - **Desventajas:** Pueden ser difíciles de entrenar debido al problema del desvanecimiento del gradiente.

Métodos estadísticos

Clasificación completa de modelos estadísticos en PLN

II. Modelos Discriminativos

- **N-gramas:**
  - **Ventajas:** Simples y eficientes de entrenar.
  - **Desventajas:** Sufren de problemas de escasez de datos.
- **Modelos de Máxima Entropía:**
  - **Ventajas:** Flexibles y se pueden usar para una variedad de tareas.
  - **Desventajas:** Pueden ser más difíciles de entrenar que otros modelos.
- **Redes Neuronales Generativas:**
  - **Ventajas:** Capaces de generar texto de alta calidad.
  - **Desventajas:** Pueden ser costosos de entrenar y requieren grandes cantidades de datos.

Métodos estadísticos

Clasificación completa de modelos estadísticos en PLN

III. Modelos para Tareas de Clasificación

- **Naive Bayes:**
  - **Ventajas:** Simple y eficiente de entrenar.
  - **Desventajas:** Asume independencia entre las características.
- **K-Nearest Neighbors:**
  - **Ventajas:** Simple y fácil de entender.
  - **Desventajas:** Puede ser lento para grandes conjuntos de datos.
- **Random Forest:**
  - **Ventajas:** Robusto y preciso.
  - **Desventajas:** Puede ser difícil de interpretar

Métodos estadísticos

Clasificación completa de modelos estadísticos en PLN

IV. Modelos para Tareas de Regresión:

- **Regresión Lineal:**
  - **Ventajas:** Simple y fácil de entender.
  - **Desventajas:** No es adecuado para datos no lineales.
- **Regresión Logística:**
  - **Ventajas:** Adecuado para tareas de clasificación binaria.
  - **Desventajas:** No es adecuado para tareas de clasificación multiclase.
- **Redes Neuronales Profundas:**
  - **Ventajas:** Capaces de aprender relaciones complejas entre las variables.
  - **Desventajas:** Pueden ser costosos de entrenar y requieren grandes cantidades de datos.

Métodos estadísticos

Clasificación completa de modelos estadísticos en PLN

V. Modelos para Tareas de Traducción:

- **Traducción Automática Estadística:**
  - **Ventajas:** Se basa en modelos estadísticos para traducir texto.
  - **Desventajas:** No es tan precisa como la traducción automática neuronal.
- **Modelos de Atención:**
  - **Ventajas:** Prestan atención a diferentes partes de una secuencia de entrada para generar una salida.
  - **Desventajas:** Pueden ser más complejos de entrenar que otros modelos.
- **Modelos de Transformadores:**
  - **Ventajas:** Muy adecuados en traducción automática.
  - **Desventajas:** Pueden ser costosos de entrenar y requieren grandes cantidades de datos.

Métodos estadísticos

Clasificación completa de modelos estadísticos en PLN

VI. Otros Modelos:

- **Modelos de lenguaje basados en la atención:** Prestan atención a diferentes partes de una secuencia de entrada para generar una salida.
- **Modelos de lenguaje basados en la memoria:** Almacenan y recuerdan información de entradas anteriores para generar una salida.

Aprendizaje Automatizado

- Algoritmos o modelos que mejoran su comportamiento con la experiencia.
- Dos formas de adquirir experiencia:
  - A partir de ejemplos suministrados por un usuario (un conjunto de ejemplos clasificados o etiquetados). APRENDIZAJE SUPERVISADO.
  - Mediante exploración autónoma (ej. software que aprende a jugar al ajedrez mediante la realización de miles de partidas contra sí mismo). APRENDIZAJE NO SUPERVISADO.

Tipos de Aprendizaje

- Aprendizaje inductivo.
  - Datos de entrada específicos: un usuario provee un subconjunto de todas las posibles situaciones.
  - Datos de salida generales: regla o modelo que puede ser aplicada a cualquier situación.
- Aprendizaje deductivo.
  - Se basa en una especialización.
- Aprendizaje por refuerzo.
  - Sistemas que aprenden mediante prueba y error.
  - Exploración autónoma para inferir reglas de comportamiento.

Aprendizaje Inductivo

- El objetivo es generar un modelo a partir de ejemplos.
- El conjunto de ejemplos usados se llama conjunto de entrenamiento.
- Cuatro elementos fundamentales: hipótesis (modelo resultante), instancias, atributos y clases.

Definiciones

- Clase: el atributo que debe ser deducido a partir de los demás.
- Atributo: cada una de las propiedades que se miden (observan) de un ejemplo.
- Instancia: cada uno de los ejemplos.
- Resultado: modelo que se infiere a partir de los ejemplos (también llamado hipótesis).

Ejemplo

EJEMPLO: Modelado de la probabilidad de fallo de una máquina.

- Clases: la máquina fallará / la máquina no fallará.
- Atributos:
  - Temperatura.
  - Nivel de vibraciones.
  - Horas de funcionamiento.
  - Meses desde la última revisión.
- Instancias: ejemplos pasados (situaciones conocidas). [Temp = alta, Nivel vibrac. = bajo, horas = 800, meses = 2, fallo = SÍ]
- Resultado: relación entre las medidas y la clase resultante.
  - SI *nivel_vibraciones = alto* Y *temp = alta* ENTONCES ***fallará***.

Atributos

Hay múltiples **tipos** de atributos:

- Real: puede tomar cualquier valor dentro de un cierto rango. Ej. temperatura como un número real [grados].
- Discreto: Ej. horas de funcionamiento como un número natural.
- Categórico: Ej. color como {azul, rojo, amarillo}
  - Se puede pensar como 'discreto no ordenado'.

Resultados

- Las hipótesis se pueden expresar de diversas formas:
  - Árboles de decisión.
  - Listas de reglas.
  - Redes neuronales.
  - Modelos bayesianos o probabilísticos.
  - Etc.

ARBOLES DE DECISIÓN

Métodos estadísticos: Arboles de Decisión

MOTIVACIÓN

#	Panorama	Temperatura	Humedad	Viento	Juega Tenis
1	Soleado	Cálida	Alta	Débil	No
2	Soleado	Cálida	Alta	Fuerte	No
3	Nublado	Cálida	Alta	Débil	Sí
4	Lluvioso	Templada	Alta	Débil	Sí
5	Lluvioso	Fría	Normal	Débil	Sí
6	Lluvioso	Fría	Normal	Fuerte	No
7	Nublado	Fría	Normal	Fuerte	Sí
8	Soleado	Templada	Alta	Débil	No
9	Soleado	Fría	Normal	Débil	Sí
10	Lluvioso	Templada	Normal	Débil	Sí
11	Soleado	Templada	Normal	Fuerte	Sí
12	Nublado	Templada	Alta	Fuerte	Sí
13	Nublado	Cálida	Normal	Débil	Sí
14	Lluvioso	Templada	Alta	Fuerte	No
15	Soleado	Cálida	Normal	Fuerte	?

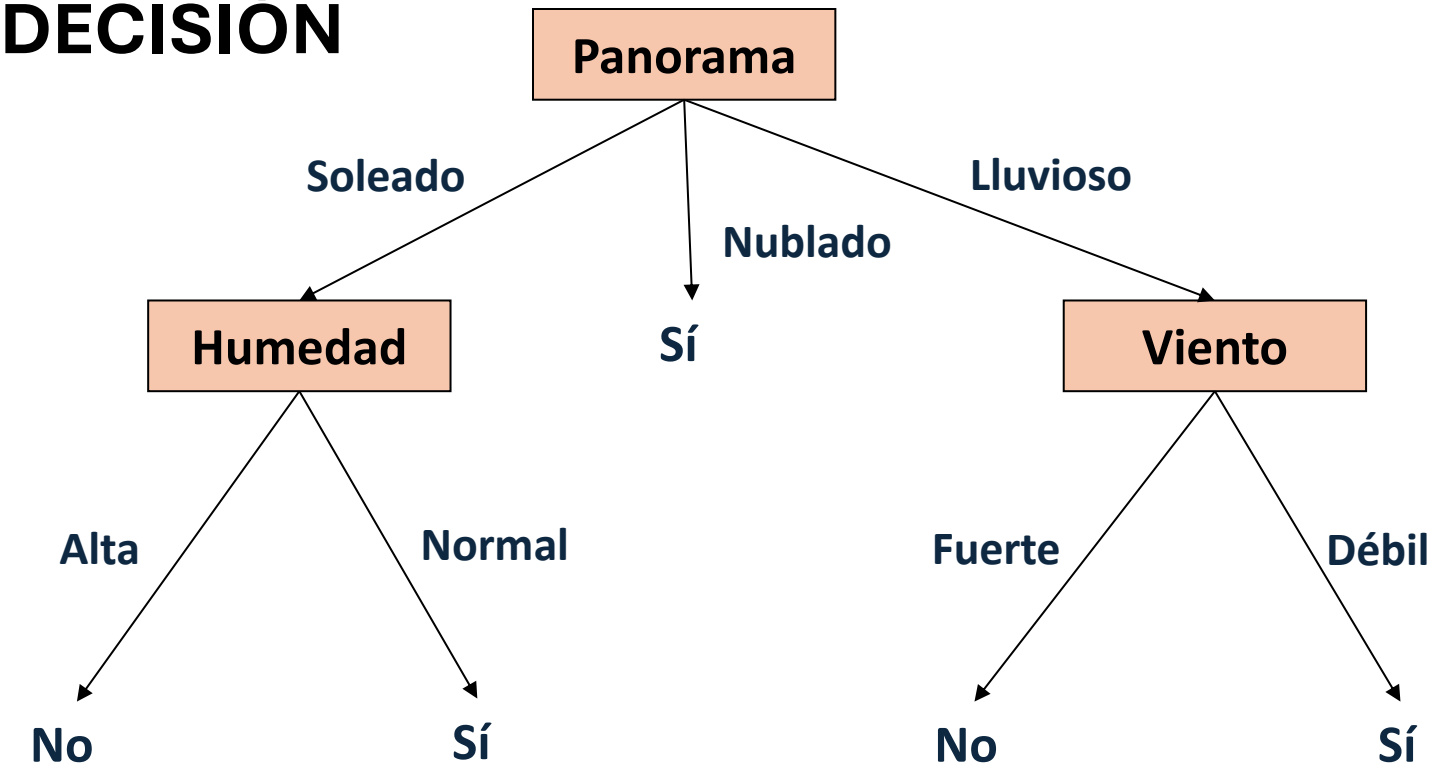
Métodos estadísticos: Arboles de Decisión

INTUICIÓN

- Los árboles de decisión (para clasificación)
- Clasifican las instancias ordenándolas según los valores de los atributos de éstas a partir de un nodo raíz hasta algún nodo hoja
- Cada nodo del árbol especifica una prueba sobre el valor de algún atributo
- Cada rama descendiente de este nodo se corresponde con uno de los posibles valores para dicho atributo

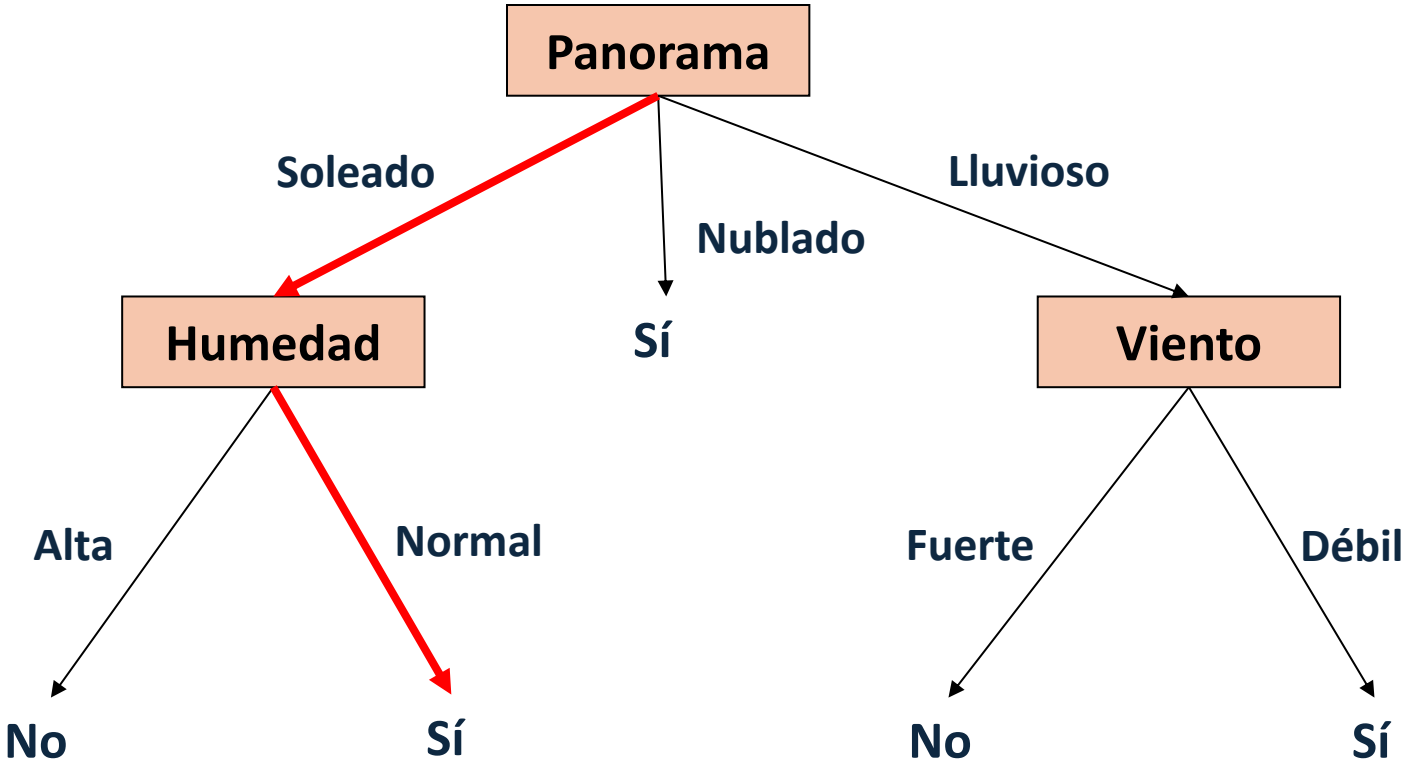
Métodos estadísticos: Arboles de Decisión

ARBOL DE DECISIÓN



Métodos estadísticos: Arboles de Decisión

INTUICIÓN

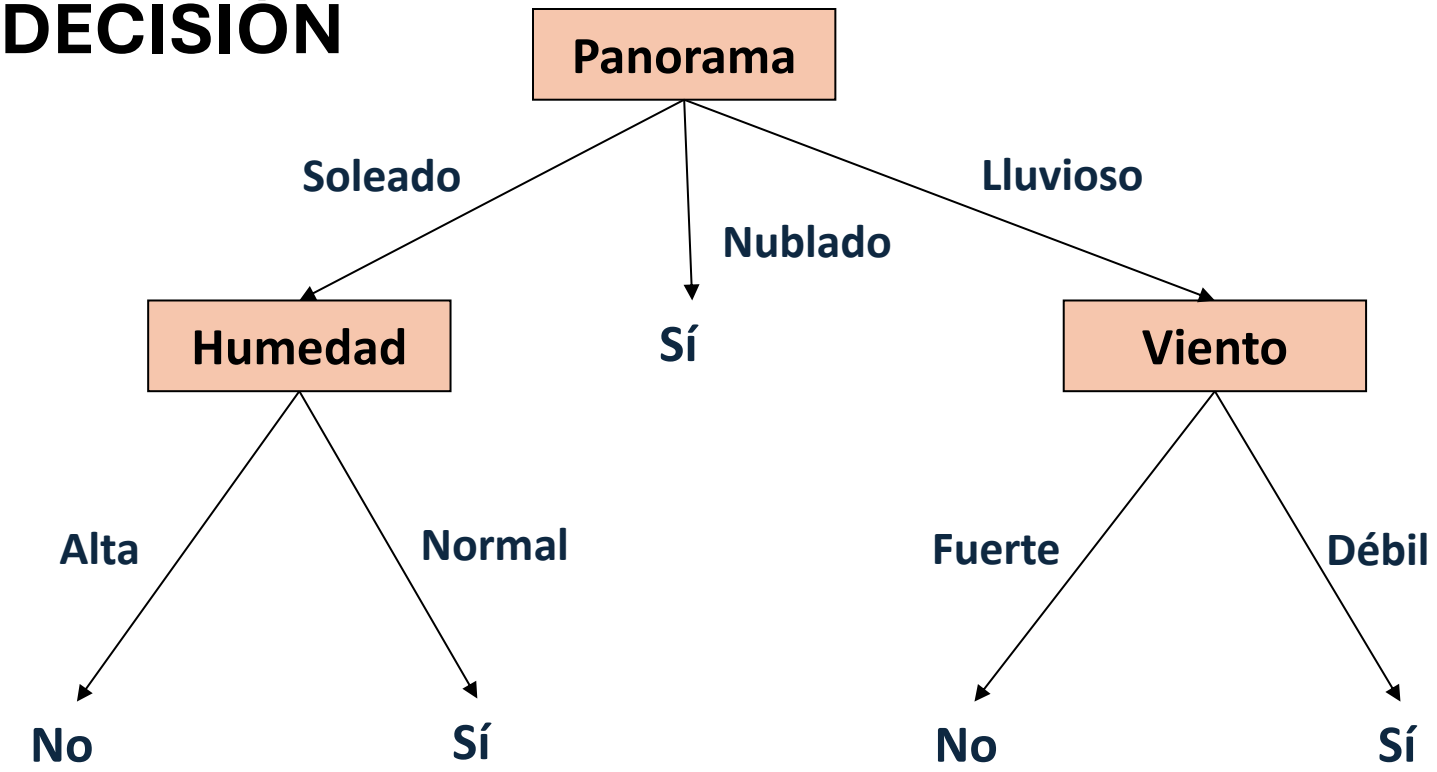


Clasificar: <Panorama=Soleado**, Temperatura=**Cálida, **Humedad=**Normal**, Viento=**Fuerte**>**

Notar que a lo largo de una rama NO se hicieron pruebas sobre el valor de todos los atributos.

Métodos estadísticos: Arboles de Decisión

ARBOL DE DECISIÓN



¿Cómo construir el árbol de manera automática?

Métodos estadísticos: Arboles de Decisión

INTUICIÓN

- **¿Cómo construir el árbol de manera automática?**
  - ¿Cómo seleccionar el nodo raíz?
  - ¿Cómo seleccionar el atributo sobre el que se decidirá en cada nodo ?

Métodos estadísticos: Arboles de Decisión

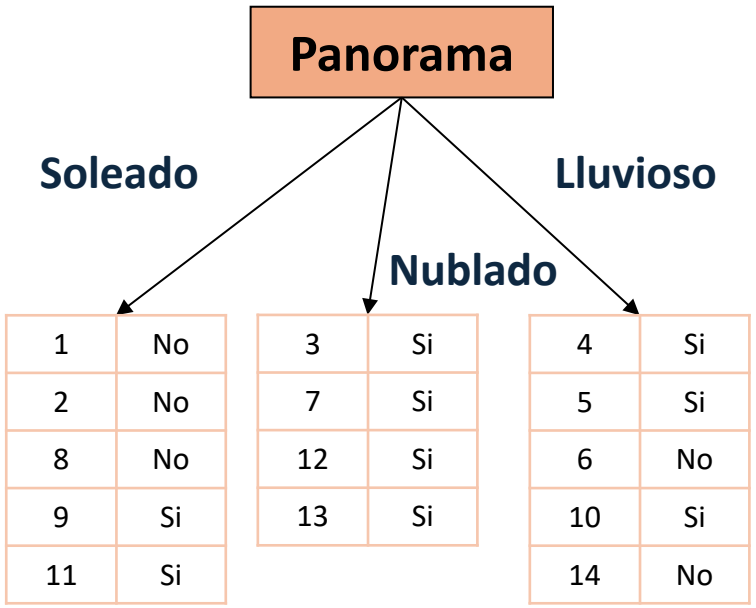
ALGORITMOS

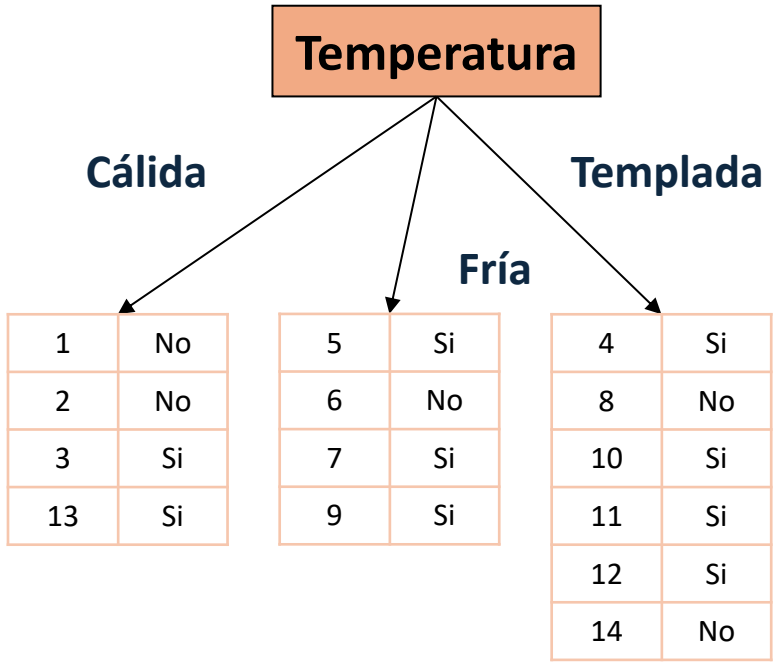
- ID3
- Decision Stump
- C4.5
- CART
- CHAID

Métodos estadísticos: Arboles de Decisión

ALGORITMO ID3

#	Panorama	Temperatura	Humedad	Viento	Juega Tenis
1	Soleado	Cálida	Alta	Débil	No
2	Soleado	Cálida	Alta	Fuerte	No
3	Nublado	Cálida	Alta	Débil	Sí
4	Lluvioso	Templada	Alta	Débil	Sí
5	Lluvioso	Fría	Normal	Débil	Sí
6	Lluvioso	Fría	Normal	Fuerte	No
7	Nublado	Fría	Normal	Fuerte	Sí
8	Soleado	Templada	Alta	Débil	No
9	Soleado	Fría	Normal	Débil	Sí
10	Lluvioso	Templada	Normal	Débil	Sí
11	Soleado	Templada	Normal	Fuerte	Sí
12	Nublado	Templada	Alta	Fuerte	Sí
13	Nublado	Cálida	Normal	Débil	Sí
14	Lluvioso	Templada	Alta	Fuerte	No





Panorama={*Soleado, Nublado, Lluvioso*}
Temperatura={*Cálida, Templada, Fría*}
Humedad={*Alta, Normal*}
Viento={*Fuerte, Débil*}

¿Qué rasgo considera "mejor"?

ALGORITMO ID3

Criterios para seleccionar atributos

- **Entropía**:
  - dado S: Conjunto de elementos conteniendo ejemplos positivos y negativos (instancias pertenecientes a dos clases), se define su **entropía** relativa a la clasificación booleana como:

$$E(S) = -p_+ log_2 p_+ - p_- log_2 p_-$$

  - Donde p_+ y p_- son la proporción de ejemplos positivos y negativos respectivamente.
- Cuando existen más de dos clases:

$$E(S) = \sum_{i=1}^{c} -p_i log_2 p_i$$

ALGORITMO ID3

Criterios para seleccionar atributos

- **Ganancia de Información:** Dados un conjunto S y un atributo A

$$Gain(S, A) = E(S) - \sum_{v \in Valores(A)} \frac{|S_v|}{|S|} E(S_v)$$

  - donde:
    - $Valores(A)$ es el conjunto de valores posibles del atributo A
    - $S_v = \{s \in S | A(s) = v\}$
    - $A(s)$= valor del atributo A en la instancia s

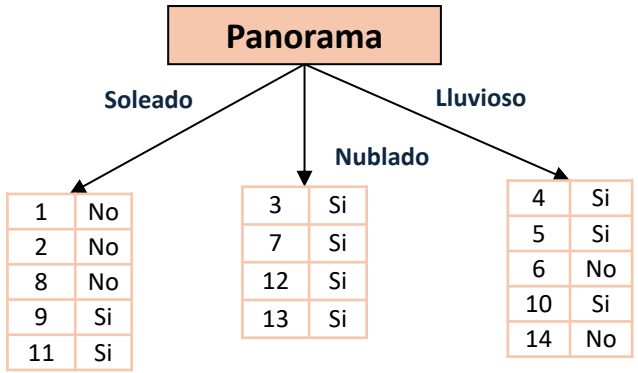
Métodos estadísticos: Arboles de Decisión

ALGORITMO ID3

Criterios para seleccionar atributos

- **Ganancia de Información**: Ejemplo

#	Panorama	Temperatura	Humedad	Viento	Juega Tenis
1	Soleado	Cálida	Alta	Débil	No
2	Soleado	Cálida	Alta	Fuerte	No
3	Nublado	Cálida	Alta	Débil	Sí
4	Lluvioso	Templada	Alta	Débil	Sí
5	Lluvioso	Fría	Normal	Débil	Sí
6	Lluvioso	Fría	Normal	Fuerte	No
7	Nublado	Fría	Normal	Fuerte	Sí
8	Soleado	Templada	Alta	Débil	No
9	Soleado	Fría	Normal	Débil	Sí
10	Lluvioso	Templada	Normal	Débil	Sí
11	Soleado	Templada	Normal	Fuerte	Sí
12	Nublado	Templada	Alta	Fuerte	Sí
13	Nublado	Cálida	Normal	Débil	Sí
14	Lluvioso	Templada	Alta	Fuerte	No



$$Gain(S, Panorama) = E(S) - \sum_{v \in Valores(A)} \frac{|S_v|}{|S|} E(S_v)$$

$S = [1,2,3,4,5,6,7,8,9,10,11,12,13,14]$

$S_{A=Soleado} = [1,2,8,9,11]$ $S_{A=Nublado} = [3,7,12,13]$

$S_{A=Lluvioso} = [4,5,6,10,14]$

$$Gain(S, Panorama) = E(S) - \left(\frac{|S_{A=Soleado}|}{|S|} E(S_{A=Soleado}) + \frac{|S_{A=Nublado}|}{|S|} E(S_{A=Nublado}) + \frac{|S_{A=Lluvioso}|}{|S|} E(S_{A=Lluvioso}) \right)$$

$E(S) = -\frac{9}{14}\log_2(\frac{9}{14}) - \frac{5}{14}\log_2(\frac{5}{14}) = 0.940$

$E(S_{A=Soleado}) = -\frac{2}{5}\log_2(\frac{2}{5}) - \frac{3}{5}\log_2(\frac{3}{5}) = 0.970$

$E(S_{A=Lluvioso}) = 0.970$ $E(S_{A=Nublado}) = 0.000$

$$Gain(S, Panorama) = 0.940 - \left(\frac{5}{14}0.970 + \frac{4}{14}0.000 + \frac{5}{14}0.970\right)$$

$$Gain(S, Panorama) = 0.940 - \left(\frac{5}{14}0.970 + \frac{4}{14}0.000 + \frac{5}{14}0.970\right)$$

$$Gain(S, Panorama) = 0.940 - (0.346 + 0.000 + 0.346) = 0.248$$

Métodos estadísticos: Arboles de Decisión

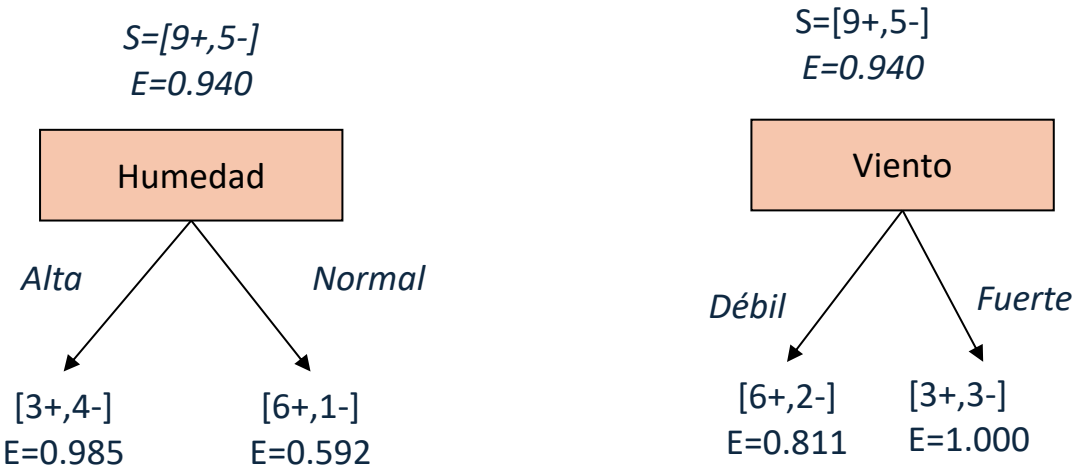
ALGORITMO ID3

Criterios para seleccionar atributos

- **Ganancia de Información**: Ejemplo

#	Panorama	Temperatura	Humedad	Viento	Juega Tenis
1	Soleado	Cálida	Alta	Débil	No
2	Soleado	Cálida	Alta	Fuerte	No
3	Nublado	Cálida	Alta	Débil	Sí
4	Lluvioso	Templada	Alta	Débil	Sí
5	Lluvioso	Fría	Normal	Débil	Sí
6	Lluvioso	Fría	Normal	Fuerte	No
7	Nublado	Fría	Normal	Fuerte	Sí
8	Soleado	Templada	Alta	Débil	No
9	Soleado	Fría	Normal	Débil	Sí
10	Lluvioso	Templada	Normal	Débil	Sí
11	Soleado	Templada	Normal	Fuerte	Sí
12	Nublado	Templada	Alta	Fuerte	Sí
13	Nublado	Cálida	Normal	Débil	Sí
14	Lluvioso	Templada	Alta	Fuerte	No

$$Gain(S,A) = E(S) - \sum_{v \in Valores(A)} \frac{|S_v|}{|S|} E(S_v)$$



$$Gain(S, Humedad) = 0.940 - \left(\frac{7}{14}0.985 + \frac{7}{14}0.5920\right) = 0.151$$

$$Gain(S, Viento) = 0.940 - \left(\frac{8}{14}0.811 + \frac{6}{14}1.000\right) = 0.048$$

$$\boldsymbol{Gain(S, Panorama) = 0.248}$$
$$Gain(S, Humedad) = 0.151$$
$$Gain(S, Viento) = 0.048$$

Panorama mayor valor -> nodo inicial

ALGORITMO ID3

Comienzo Algoritmo ID3 (E: Ejemplos, A: Atributos):

1 – Crear el nodo raíz.

1.1 - Si todas las instancias en Ejemplos pertenecen a una misma clase devolver un árbol con un solo nodo etiquetado con la clase de los ejemplos.

2 – Si Atributos es vacío, devolver un árbol con un único nodo etiquetado con la clase más probable en Ejemplos.

3 – De otro modo:

…….. (continua)

ALGORITMO ID3

3 – De otro modo:

3.1- A=Atributo que mejor[1] clasifica las instancias en Ejemplos.

3.2- Etiquetar el nodo raíz con A.

3.3- Para cada valor (vi) posible del atributo A:

3.3.1-$Ejemplos_{vi}$ = instancias en Ejemplos cuyo valor para el atributo A es vi.

3.3.2- Si $Ejemplos_{vi}$ es vacío:

- Adicionar a continuación del nodo actual un nodo hoja etiquetándolo con la clase mayoritariamente representada en Ejemplos y la arista con vi.

3.3.3- Si $Ejemplos_{vi}$ no es vacío:

- Adicionar a continuación del nodo actual un subárbol construido llamando recursivamente a ID3($Ejemplos_{vi}$,{{Atributos}-A}), etiquetar la arista con vi.

Fin.

1- En este caso el mejor atributo es el de mayor ganancia de información.

ALGORITMO ID3
Otros criterios para seleccionar atributos

$$SplitInformation(S,A) = \sum_{v \in Valores(A)} \frac{|S_v|}{|S|} log_2 \frac{|S_v|}{|S|}$$

- Es la entropía de S respecto a los valores del atributo

- Penaliza atributos con muchos valores uniformemente distribuidos

$$GainRatio(S,A) = \frac{Gain(S,A)}{SplitInformation(S,A)}$$

- Incluye una penalización a los atributos que tienden a dividir los datos de manera uniforme

- Como inconveniente, toma valores muy altos o se indefine cuando un atributo toma el mismo valor para casi todas las instancias, $|S_v| \approx |S|$

ALGORITMO ID3

Entropía Condicional (para realizar una partición binaria)

$$E(S|A = v) = \frac{|S_v|}{|S|} E(S_v) + \frac{|S_{\neq v}|}{|S|} E(S_{\neq v})$$

donde:

- S: Conjunto de elementos conteniendo ejemplos positivos y negativos (instancias pertenecientes a dos clases)

- $v \in Valores(A)$ es el conjunto de valores posibles del atributo A
- $A(s) =$ valor del atributo A en la instancia s
- $S_v = \{s \in S | A(s) = v\}$
- $S_{\neq v} = \{s \in S | A(s) \neq v\}$

Métodos estadísticos: Arboles de Decisión

ALGORITMO Decision Stump

- Son árboles de decisión que particionan los datos empleando un único atributo (tiene un solo nivel)
- Para **atributos discretos nominales**, el árbol en general contiene un único nodo que separa las instancias que tienen valor v para el atributo, aquellas que tienen otro valor y otras cuyo valor es desconocido.
- Para **atributos discretos ordinales** o **numéricos**, el nodo separa en instancias cuyo valor es menor a un umbral y aquellas con valor mayor. Puede también ser más complejo, al emplear varios umbrales.
- A pesar de su simpleza, brinda buenos resultados en muchos problemas aunque en general se emplea como *weak classifier* en los algoritmos de boosting, como Adaboost.

ALGORITMO CART (Classification and Regression Tree)

- Crea árboles binarios (en su formulación básica)

- Sea M el número de ejemplos en el conjunto de entrenamiento, A un atributos con valores $a_1, a_2, \ldots, a_M$ (valores del atributo en cada ejemplo) y m_i^A el valor del atributo A en la instancia m_i:
  - Para **atributos discretos nominales**, cada posible sub-conjunto C_k^A de los valores del atributo origina un particionamiento de las instancias en dos conjuntos de acuerdo a si $m_i^A \in C_k^A$ o $m_i^A \notin C_k^A$.
  - Para **atributos discretos ordinales**, cada posible valor del atributo genera un particionamiento de acuerdo a $m_i^A \leq a_k$ o $m_i^A > a_k$
  - Para **atributos continuos**, cada posible valor del atributo genera un particionamiento de acuerdo a $m_i^A \leq a_k$ o $m_i^A > a_k$

ALGORITMO CART (Classification and Regression Tree)

Criterios para seleccionar atributos

Gini Impurity

Dados un conjunto S y un atributo A

$$i(S) = 1 - \sum_j P^2(w_j)$$

donde:

-$P(w_j)$ es la frecuencia de la clase w_j en S.

Miss classification Impurity

$$i(S) = 1 - max_j\{P(w_j)\}$$

Impurity Decrease

Se define para un posible particionamiento SP que genera ramas $sp_1, sp_2, \dots, sp_n$

$$i(S, SP) = i(S) - \sum_{spi} \frac{|S_{spi}|}{|S|} i(S_{sp_i})$$

donde:

- S_{sp_i} son las instancias de S agrupadas en la rama sp_i

Métodos estadísticos: Arboles de Decisión

Elementos a tener en cuenta

¿Hasta que punto hacer crecer el árbol?

¿Cómo tratar atributos numéricos?

¿Cómo tratar instancias que tienen algún atributo con valor desconocido?

¿Qué efecto tienen los atributos que pueden tomar muchos valores?

¿Cómo evitar el sobreajuste?

Métodos estadísticos: Arboles de Decisión

Atributos con valor desconocido

#	Panorama	Temperatura	Humedad	Viento	Juega Tenis
1	Soleado	Cálida	Alta	Débil	No
2	Soleado	Cálida	Alta	Fuerte	No
8	Soleado	Templada	Alta	Débil	No
9	Soleado	Fría	Normal	Débil	Sí
11	Soleado	Templada	?	Fuerte	Sí

Opción 1: Asignar el valor más común de las instancias en el nodo:

Humedad=Alta. (3/4)

Métodos estadísticos: Arboles de Decisión

Atributos con valor desconocido

#	Panorama	Temperatura	Humedad	Viento	Juega Tenis
1	Soleado	Cálida	Alta	Débil	No
2	Soleado	Cálida	Alta	Fuerte	No
8	Soleado	Templada	Alta	Débil	No
9	Soleado	Fría	Normal	Débil	Sí
11	Soleado	Templada	?	Fuerte	Sí

Opción 2: Asignar el valor más común de las instancias de la misma clase en el nodo:

Humedad=Normal (1/1)

Métodos estadísticos: Arboles de Decisión

Atributos con valor desconocido

- Crear tantas instancias como posibles valores v_i tome el atributo

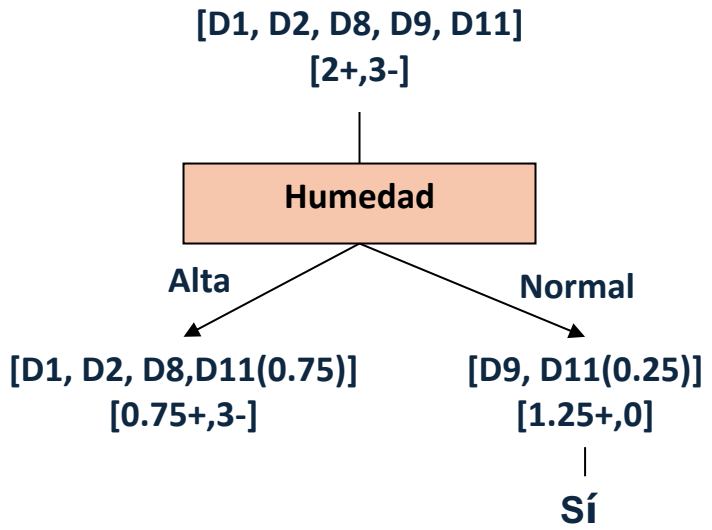
- Asignar a cada instancia un peso p_i de acuerdo a la probabilidad de que el atributo tome el valor v_i

- Al contar la cantidad de instancias en el nodo, la instancia contará ponderada de acuerdo a p_i (también al clasificar)

$$P_{Humedad=Alta} = 0.75$$

$$P_{Humedad=Normal} = 0.25$$

#	Panorama	Temperatura	Humedad	Viento	Juega Tenis
1	Soleado	Cálida	Alta	Débil	No
2	Soleado	Cálida	Alta	Fuerte	No
8	Soleado	Templada	Alta	Débil	No
9	Soleado	Fría	Normal	Débil	Sí
11	Soleado	Templada	?	Fuerte	Sí



Métodos estadísticos: Arboles de Decisión

Atributos con valor desconocido

Cálculo de $E(S_{Humedad=Alta})$

$$E(S_{Humedad=Alta}) = -p_+ log_2 p_+ - p_- log_2 p_-$$

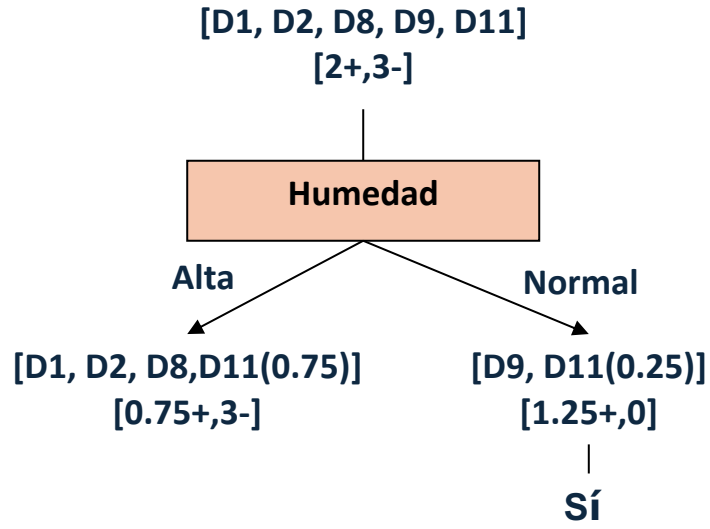
$$p_+ = \frac{0.75}{3.75} = 0.2 \quad \text{y} \quad p_- = \frac{3}{3.75} = 0.8$$

$$E(S_{Humedad=Alta}) = -0.2\ log_2 0.2 - 0.8\ log_2 0.8$$

$$E(S_{Humedad=Alta}) = 0.72$$

$$E(S_{Humedad=Normal}) = -\frac{1.25}{1.25} log_2 1 - \frac{0}{1.25} log_2 0 = 0$$

#	Panorama	Temperatura	Humedad	Viento	Juega Tenis
1	Soleado	Cálida	Alta	Débil	No
2	Soleado	Cálida	Alta	Fuerte	No
8	Soleado	Templada	Alta	Débil	No
9	Soleado	Fría	Normal	Débil	Sí
11 (0,75)	Soleado	Templada	ALTA	Fuerte	Sí
11 (0,25)	Soleado	Templada	NORMAL	Fuerte	Sí



Atributos numéricos

Dos casos:

- Rasgos numéricos
- Atributo a predecir numérico (problema de regresión)

Rasgos numéricos

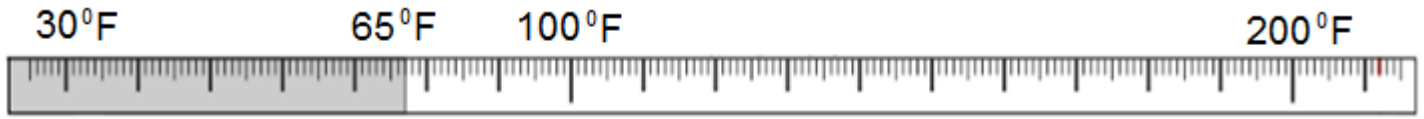
- Definir dinámicamente un nuevo atributo discreto que particione el atributo continuo en un conjunto discreto de intervalos

Ejemplo $v_i \in \{Baja, Media, Alta\}$

$$v_i = \begin{cases} Baja & si \quad t \leq 65 \\ Alta & > 65 \end{cases}$$

Temperatura (F)	Juega Tenis
72	Sí
48	No
80	Sí
40	No
60	Sí
90	No

¿Cómo seleccionar los valores para cada umbral?



Métodos estadísticos: Arboles de Decisión

Atributos numéricos

Dos casos:

- Rasgos numéricos
- Atributo a predecir numérico (problema de regresión)

Rasgos numéricos

- Definir dinámicamente un nuevo atributo discreto que particione el atributo continuo en un conjunto discreto de intervalos

Ejemplo $v_i \in \{Baja, Media, Alta\}$

$$v_i = \begin{cases} Baja & si \quad t \leq 65 \\ Alta & > 65 \end{cases}$$

Temperatura (F)	Juega Tenis
72	Sí
48	No
80	Sí
40	No
60	Sí
90	No

Temperatura (F)	Juega Tenis
40	No
48	No
60	Sí
72	Sí
80	Sí
90	No

¿Cómo seleccionar los valores para cada umbral?

- Seleccionar los umbrales que produzcan mayor ganancia de información, es decir, deje lo mejor clasificadas posible a las instancias.

$$U = \frac{(48 + 60)}{2}$$

Métodos estadísticos: Arboles de Decisión

Otros criterios para seleccionar atributos

¿Qué efecto tienen los atributos que pueden tomar muchos valores?

#	Fecha	Panorama	Temperatura	Humedad	Viento	Juega Tenis
1	02/01/2016	Soleado	Cálida	Alta	Débil	No
2	09/01/206	Soleado	Cálida	Alta	Fuerte	No
3	16/01/2016	Nublado	Cálida	Alta	Débil	Sí
4	17/01/2016	Lluvioso	Templada	Alta	Débil	Sí
5	22/01/2016	Lluvioso	Fría	Normal	Débil	Sí
6	23/01/2066	Lluvioso	Fría	Normal	Fuerte	No
7	30/01/2016	Nublado	Fría	Normal	Fuerte	Sí
8	06/02/2016	Soleado	Templada	Alta	Débil	No
9	07/02/2016	Soleado	Fría	Normal	Débil	Sí
10	12/02/2016	Lluvioso	Templada	Normal	Débil	Sí
11	13/02/2016	Soleado	Templada	Normal	Fuerte	Sí
12	14/02/2016	Nublado	Templada	Alta	Fuerte	Sí
13	20/02/2016	Nublado	Cálida	Normal	Débil	Sí
14	21/02/2016	Lluvioso	Templada	Alta	Fuerte	No

$$Gain(S, Fecha) = E(S) - \sum_{v \in Valores(Fecha)} \frac{|S_v|}{|S|} E(S_v)$$

Métodos estadísticos: Arboles de Decisión

Otros criterios para seleccionar atributos

¿Qué efecto tienen los atributos que pueden tomar muchos valores?

$$Gain(S, Fecha) = E(S) - \sum_{v \in Valores(Fecha)} \frac{|S_v|}{|S|} E(S_v)$$

$$E\left(S_{Fecha=02/01/2016}\right) = 0$$

$$E\left(S_{Fecha=09/01/2016}\right) = 0$$

…

¡Fecha es el atributo con mayor GA!

En general los atributos con "muchos valores" posibles tienden a separar los datos en muchos conjuntos pequeños y a tener altos valores de GA

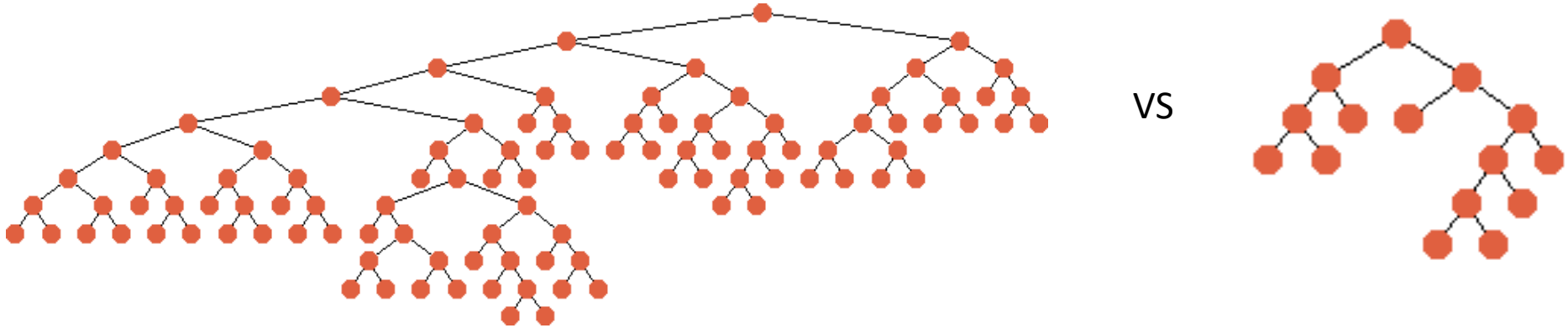
"muchos valores" $\sim Valores(A) > \frac{|S|}{3}$

#	Fecha	Juega Tenis
1	02/01/2016	No
2	09/01/206	No
3	16/01/2016	Sí
4	17/01/2016	Sí
5	22/01/2016	Sí
6	23/01/2066	No
7	30/01/2016	Sí
8	06/02/2016	No
9	07/02/2016	Sí
10	12/02/2016	Sí
11	13/02/2016	Sí
12	14/02/2016	Sí
13	20/02/2016	Sí
14	21/02/2016	No

Métodos estadísticos: Arboles de Decisión

Evitar el sobreajuste: Occam's razor

Preferir el modelo más simple que se ajuste a los datos



VS

Dos enfoques principales

- Detener el crecimiento del árbol, antes de que clasifique correctamente todos los ejemplos
- No controlar el crecimiento pero realizar una etapa de poda. En general este enfoque ha brindado mejores resultados.

Métodos estadísticos: Arboles de Decisión

Evitar el sobreajuste: Dos enfoques principales

- **Detener el crecimiento del árbol, aun cuando no clasifique bien todos los ejemplos**
  - Se ha alcanzado una profundidad máxima determinada
  - El número de instancias en un nodo es menor a un umbral deseado
  - El valor del criterio de particionamiento no sobrepasa un valor determinado

- **No controlar el crecimiento pero realizar una etapa de poda. En general este enfoque ha brindado mejores resultados**
  - *Subtree raising*
  - *Subtree replacement*

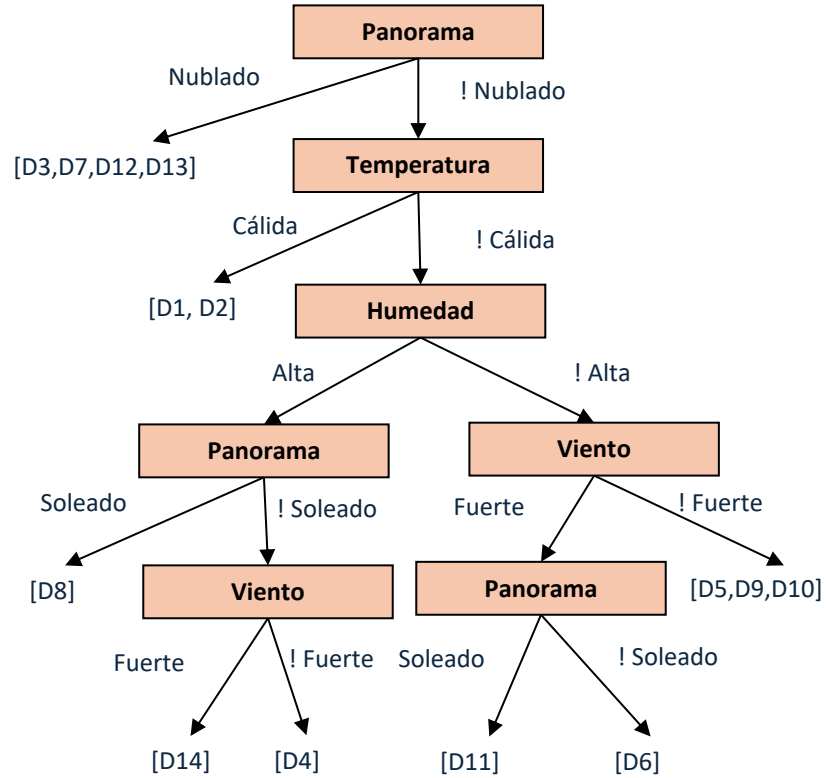
Métodos estadísticos: Arboles de Decisión

Subtree raising

Eliminar un nodo interno, reemplazándolo por uno de sus los sub-árboles

- Redistribuir las instancias del resto de los sub-árboles del nodo eliminado

Ejemplo:



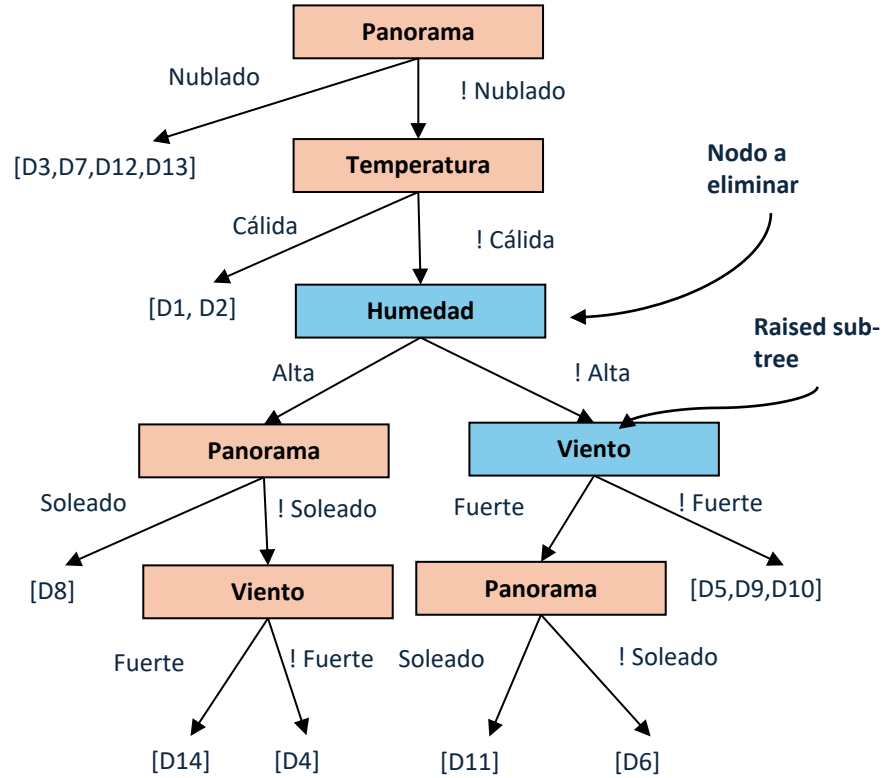
Métodos estadísticos: Arboles de Decisión

Subtree raising

Eliminar un nodo interno, reemplazándolo por uno de sus los sub-árboles

- Redistribuir las instancias del resto de los sub-árboles del nodo eliminado

Ejemplo:



¿Qué nodos eliminar?
¿Qué sub-árbol colocar en su lugar?

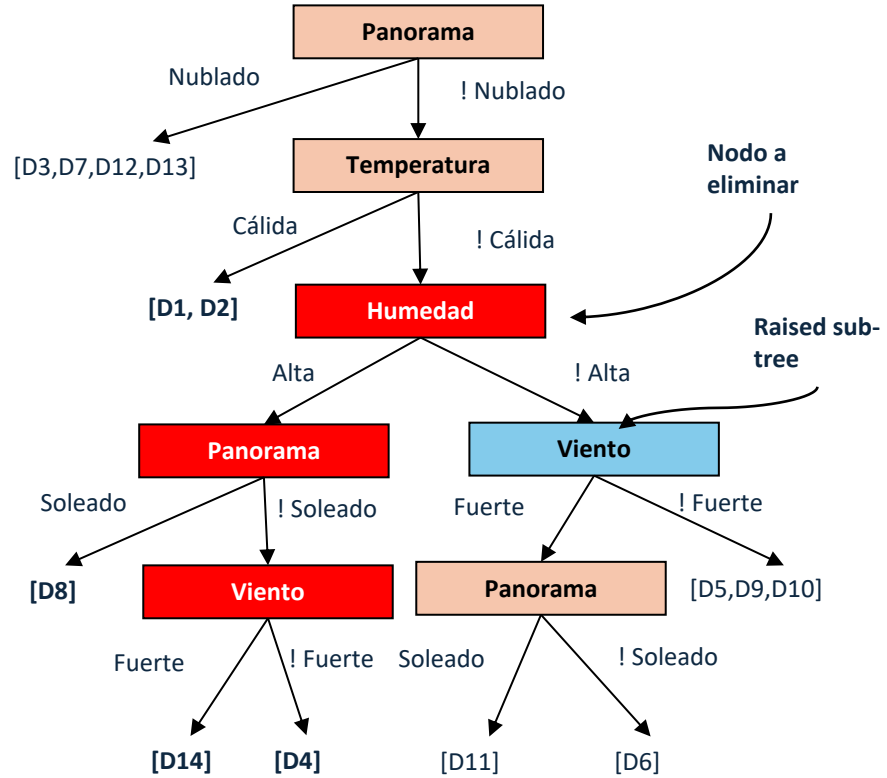
Métodos estadísticos: Arboles de Decisión

Subtree raising

Eliminar un nodo interno, reemplazándolo por uno de sus los sub-árboles

- Redistribuir las instancias del resto de los sub-árboles del nodo eliminado

Ejemplo:



¿Qué nodo eliminar?
¿Qué sub-árbol colocar en su lugar?

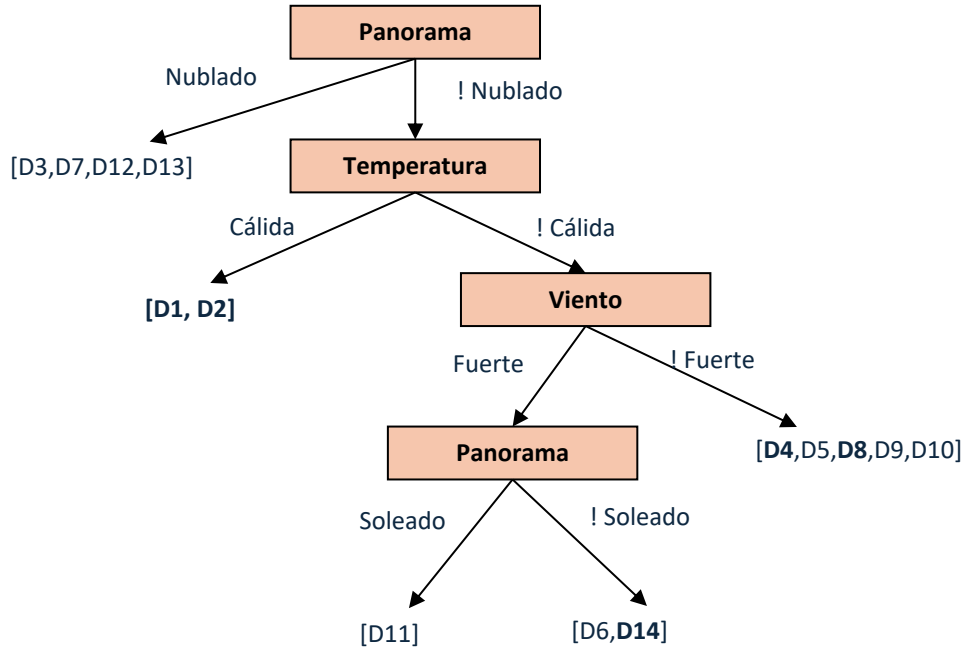
Métodos estadísticos: Arboles de Decisión

Subtree raising

Eliminar un nodo interno, reemplazándolo por uno de sus los sub-árboles

- Redistribuir las instancias del resto de los sub-árboles del nodo eliminado

Ejemplo:



¿Qué nodo eliminar?
¿Qué sub-árbol colocar en su lugar?

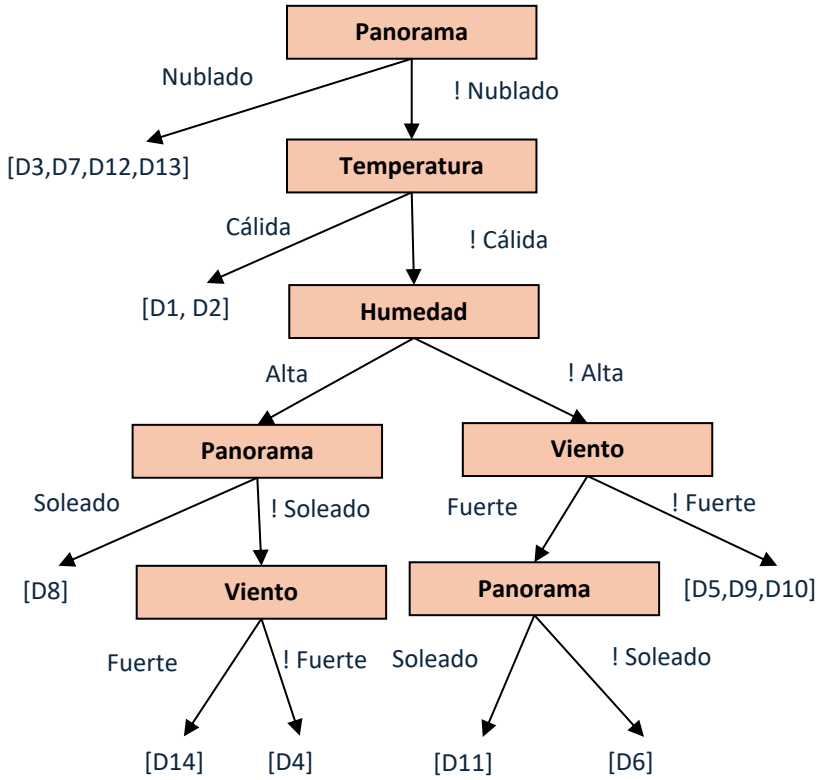
Métodos estadísticos: Arboles de Decisión

Subtree replacement

- Eliminar un nodo interno, reemplazándolo por un nodo hoja

- Redistribuir las instancias del resto de los sub-árboles del nodo eliminado y asignar la clase según el caso (ejemplo, clase mayoritaria en el nodo)

Ejemplo:



¿Qué nodo eliminar?
¿Qué sub-árbol colocar en su lugar?

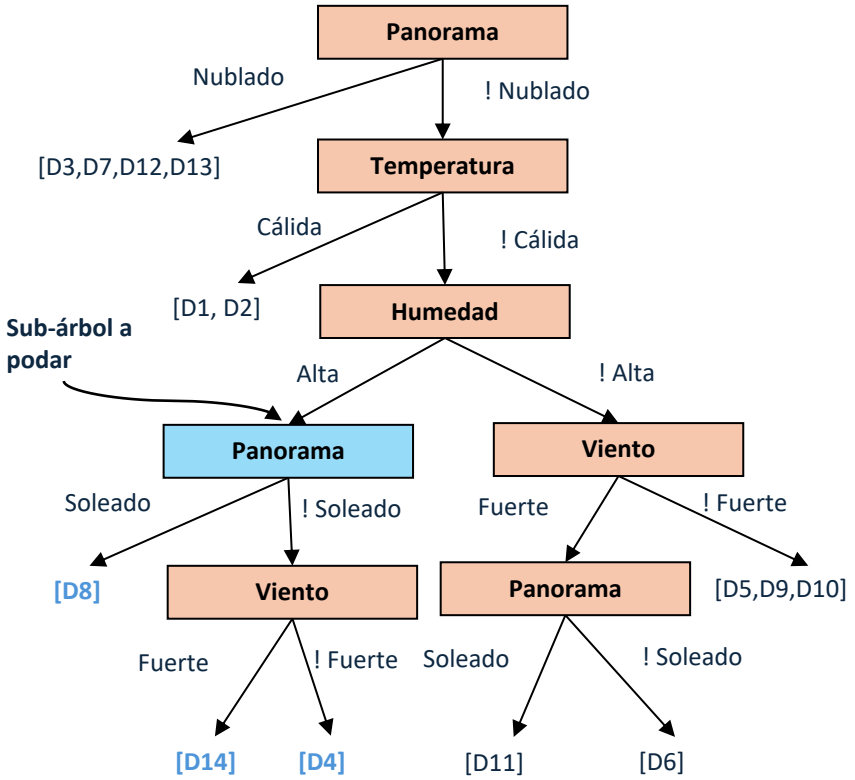
Métodos estadísticos: Arboles de Decisión

Subtree replacement

- Eliminar un nodo interno, reemplazándolo por un nodo hoja

- Redistribuir las instancias del resto de los sub-árboles del nodo eliminado y asignar la clase según el caso (ejemplo, clase mayoritaria en el nodo)

Ejemplo:



¿Qué nodo eliminar?
¿Qué sub-árbol colocar en su lugar?

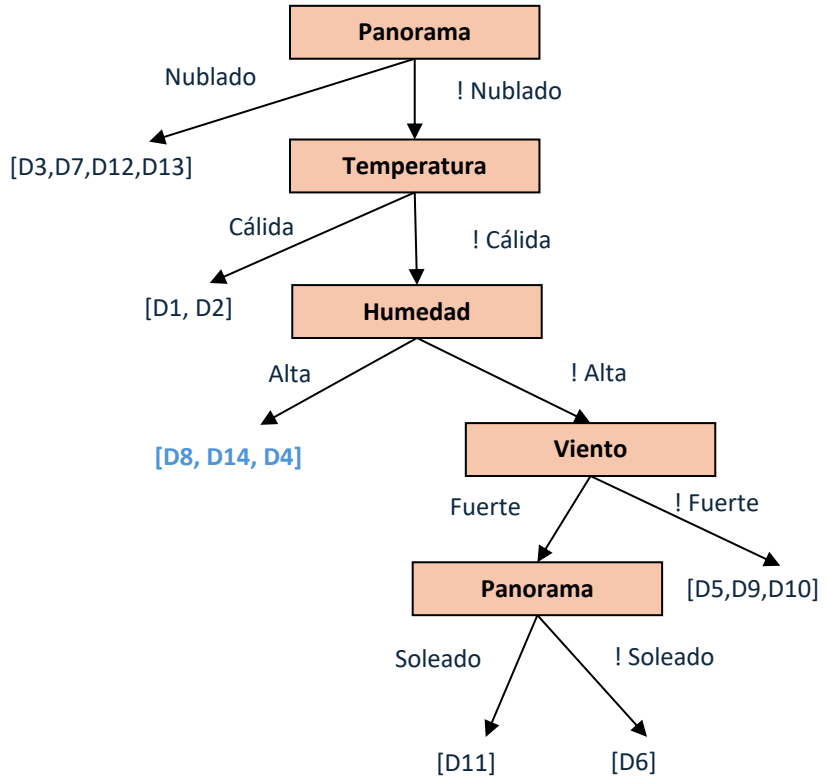
Métodos estadísticos: Arboles de Decisión

Subtree replacement

- Eliminar un nodo interno, reemplazándolo por un nodo hoja

- Redistribuir las instancias del resto de los sub-árboles del nodo eliminado y asignar la clase según el caso (ejemplo, clase mayoritaria en el nodo)

Ejemplo:



¿Qué nodo eliminar?
¿Qué sub-árbol colocar en su lugar?

Métodos estadísticos: Arboles de Decisión

Evitar el sobreajuste. Poda

Poda basada en la Reducción del Error

Sean T, V, P conjuntos de entrenamiento, validación y prueba respectivamente:

- Considerar cada nodo como un candidato a ser podado

- La poda de un nodo significa remover el subárbol cuya raíz es el nodo seleccionado, convirtiendo al nodo en hoja

- Asignar al nodo la clase más frecuente en los ejemplos a el asociados

- Los nodos se remueven solo si el nuevo árbol se desempeña mejor en el conjunto de prueba

- Los nodos se remueven de iterativamente, seleccionado cada vez el que conduzca a la mayor disminución del error

- La poda puede continuar hasta que se cumpla algún criterio, por ejemplo que cualquier cambio conduzca a incrementos del error

Métodos estadísticos: Arboles de Decisión

Evitar el sobreajuste. Poda

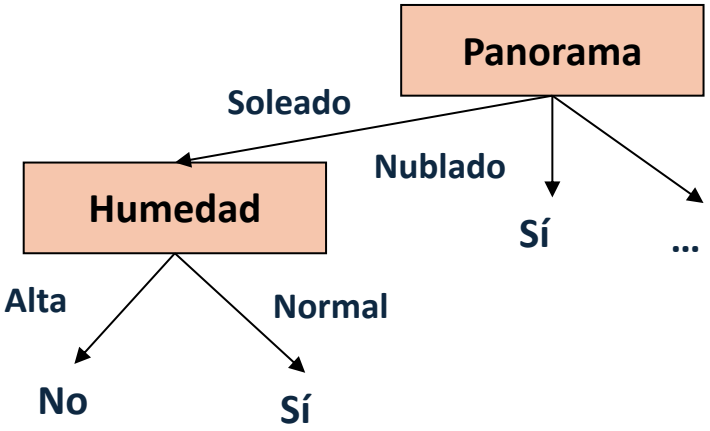
Poda basada en Reglas

- Construir el árbol de decisión a partir del conjunto de entrenamiento sin controlar el crecimiento

- Convertir el árbol en un conjunto de reglas equivalentes, creando una regla por cada camino desde la raíz hasta las hojas

- Podar (hacer más general) cada regla eliminando cualquier precondición que resulte en un aumento de la precisión

- Ordenar las reglas de acuerdo a un estimado de su precisión y considerarlas en este orden al realizar la clasificación de un ejemplo



SI Panorama=Soleado **Y** Humedad=Alta **ENTONCES** JuegaTenis=No

SI Panorama=Soleado **Y** Humedad=Normal **ENTONCES** JuegaTenis=Si

SI Panorama=Nublado **ENTONCES** JuegaTenis=Si

Métodos estadísticos

NAIVE BAYES

Métodos estadísticos: Naive Bayes

- Naive Bayes es una técnica de clasificación estadística basada en el Teorema de Bayes
- Es uno de los algoritmos de aprendizaje supervisado más sencillos
- Es un algoritmo rápido, preciso y fiable con grandes conjuntos de datos
- Modela la relación probabilística entre atributos y clase
- Modelo generativo capaz de generar datos
- Teorema de Bayes. Revisa las probabilidades cuando se pose nueva información
  - Infiere probabilidad de una ocurrencia en función al conocimiento que tiene

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

Métodos estadísticos: Naive Bayes

El teorema de **Bayes**

- Supongamos que sobre el espacio muestral **S** tenemos una partición $\mathbf{A}_i$, con $i = 1, \ldots, n$.
- Esto significa que cualquier resultado de S necesariamente debe estar en uno y solo uno de los eventos A_i

Por ejemplo, los pacientes hospitalizados en una ciudad, y la ciudad tiene cuatro hospitales, digamos los hospitales 1, 2, 3 y 4. De modo que el conjunto de pacientes hospitalizados va a estar en uno y solo uno de esos cuatro hospitales.

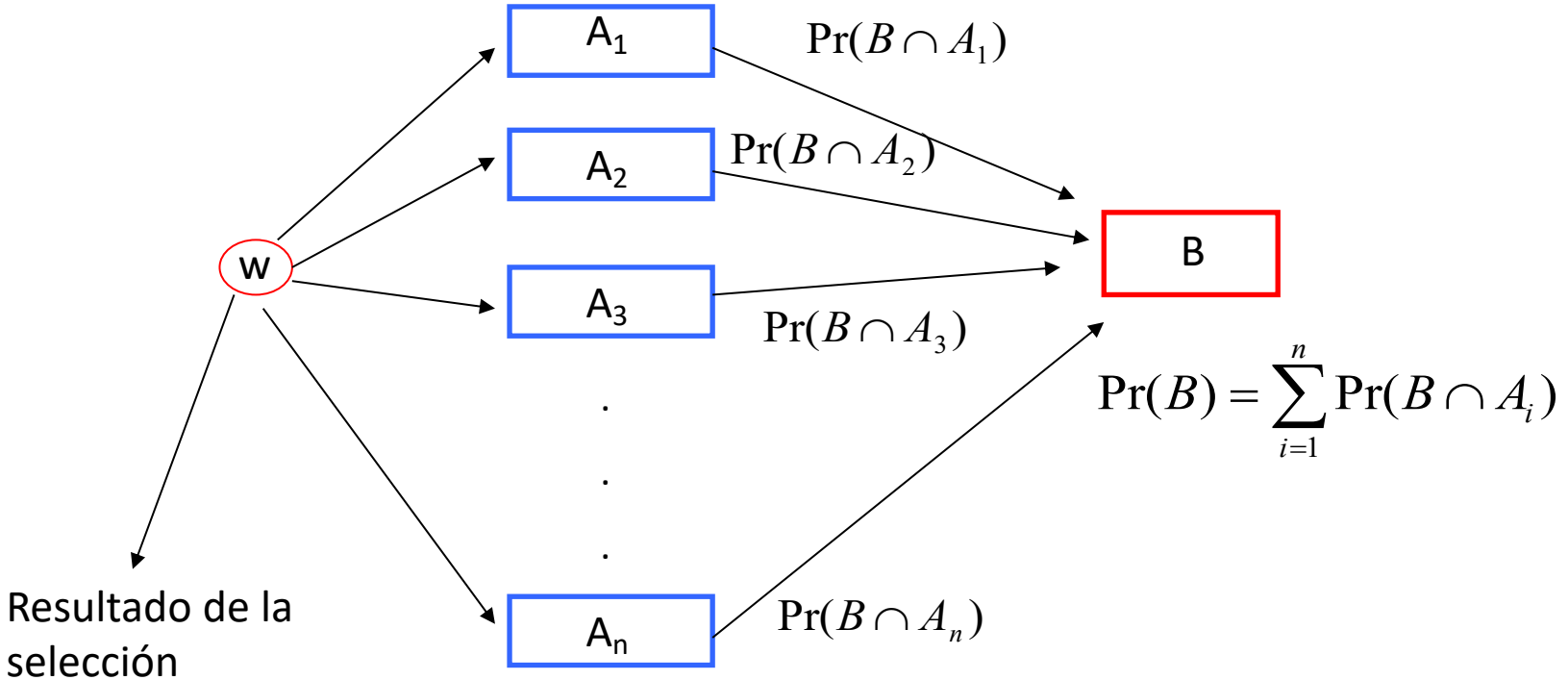
Si definimos los sucesos $\mathbf{A}_i$ como el conjunto de pacientes hospitalizados en el i-ésimo hospital, con $i = 1, 2, 3, 4$. Entonces los sucesos A_1, A_2, A_3 y A_4 constituyen una partición sobre el conjunto de todos los pacientes hospitalizados, que llamaremos **S.**

De otra forma, si seleccionamos al azar un paciente hospitalizado, entonces el paciente que elegiremos pertenecerá a uno y solo uno de los A_i.

Métodos estadísticos: Naive Bayes

El teorema de **Bayes**

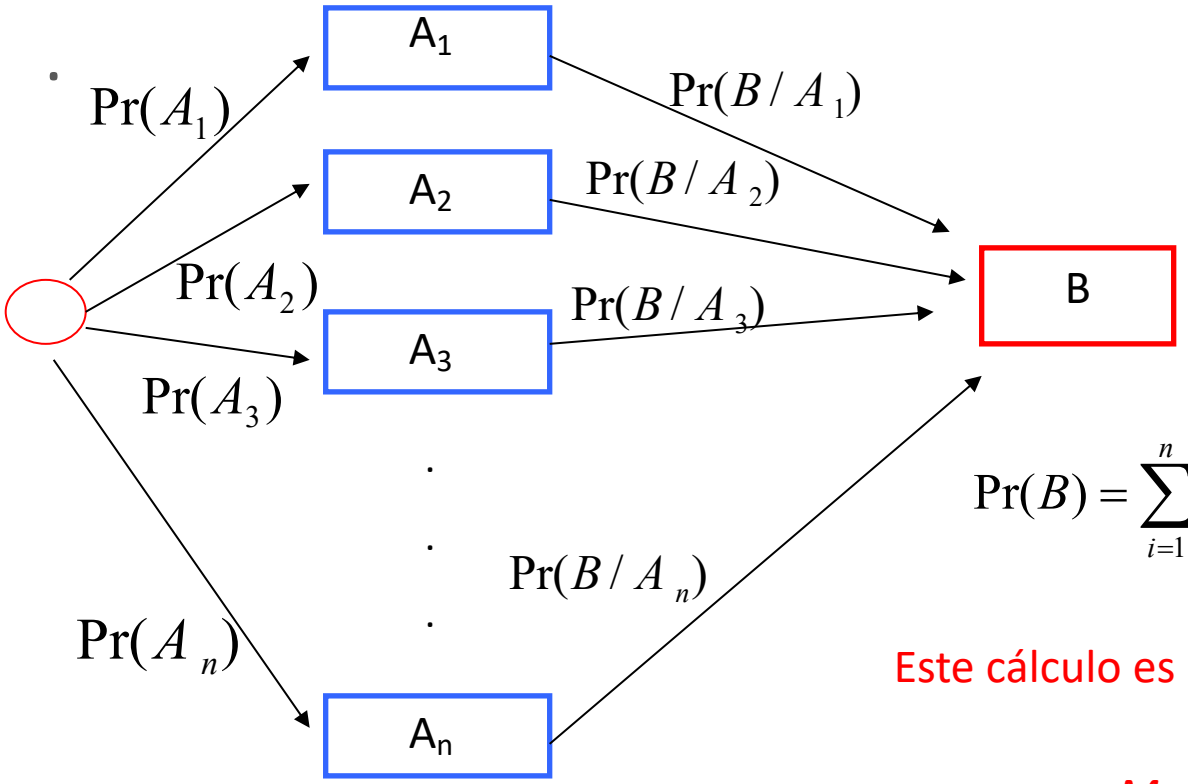
- Consideremos un suceso B, que indica una determinada propiedad de los pacientes, por ejemplo B puede ser el suceso de que el paciente seleccionado al azar tenga un diagnóstico grave.



Métodos estadísticos: Naive Bayes

El teorema de **Bayes**

- En función de las probabilidades condicionales, nos queda



$$\Pr(B) = \sum_{i=1}^{n} \Pr(B / A_i) \Pr(A_i)$$

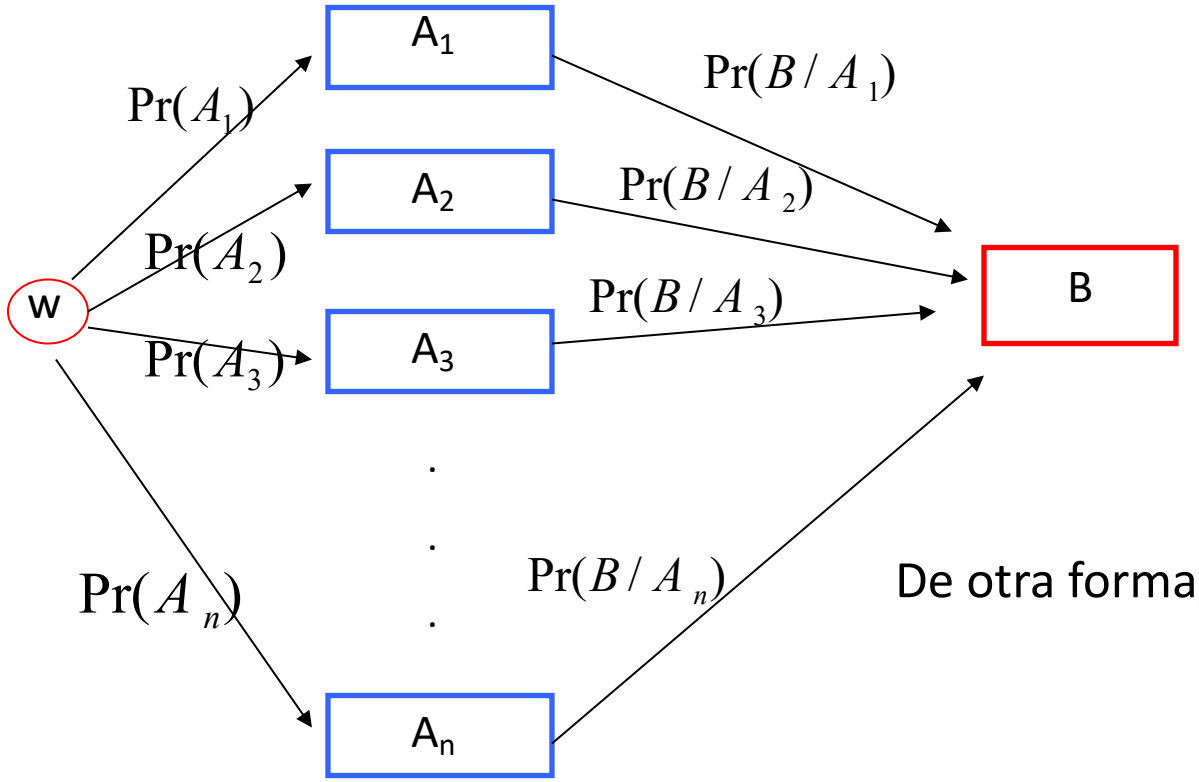
Este cálculo es para medir la incertidumbre de la ocurrencia del evento B.

Medición del futuro, representado por el evento B

Métodos estadísticos: Naive Bayes

El teorema de **Bayes**

- Supongamos que ocurre B, ¿Cuál de los sucesos A_j ha ocurrido?

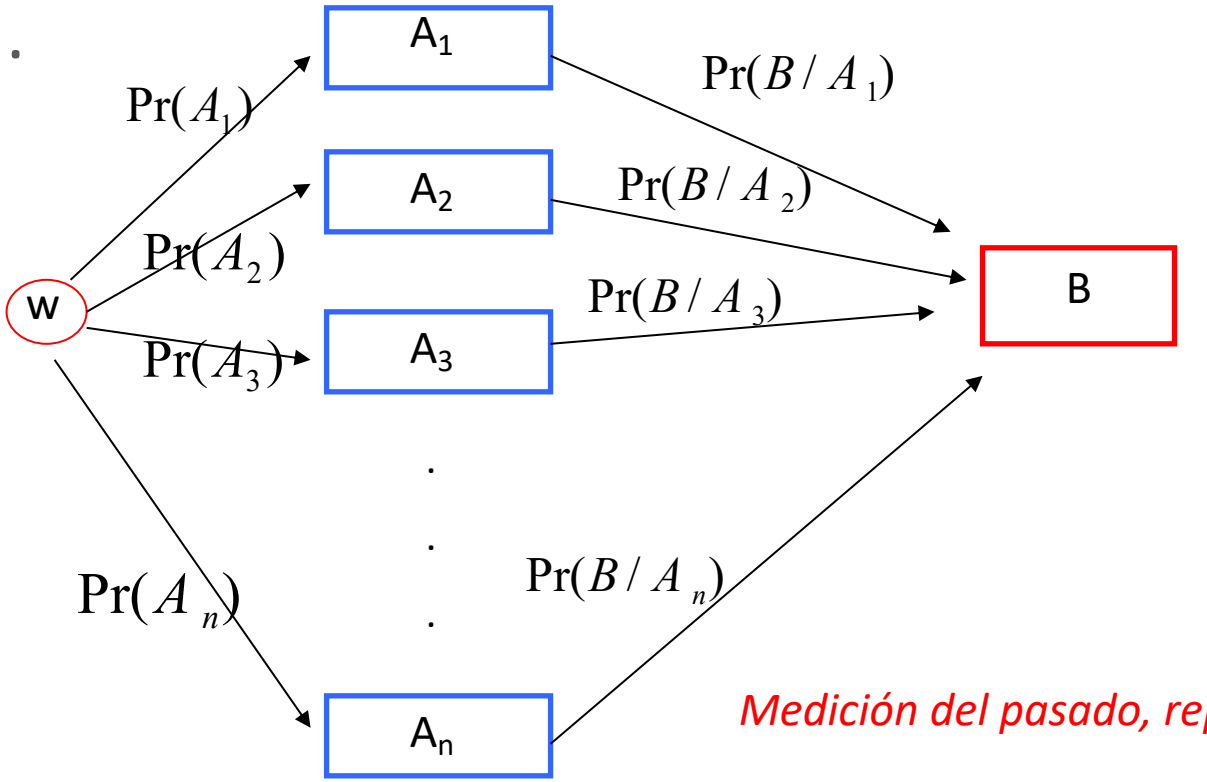


De otra forma, ¿cuál es el valor de $\Pr(A_j / B)$ con j = 1, ...n?

Métodos estadísticos: Naive Bayes

El teorema de **Bayes**

- Supongamos que ocurre B, ¿Cuál de los sucesos A_j ha ocurrido?



$$\Pr(A_j / B) = \frac{\Pr(A_j \cap B)}{\Pr(B)} = \frac{\Pr(B / A_j) \cdot \Pr(A_j)}{\Pr(B)}$$

$$\Pr(A_j / B) = \frac{\Pr(B / A_j) \cdot \Pr(A_j)}{\sum_{i=1}^{n} P(B / A_i) \cdot \Pr(A_i)}$$

Medición del pasado, representado por el evento A_j

Métodos estadísticos: Naive Bayes

Clasificador bayesiano

$$P(C|x) = \frac{P(x|C) * P(C)}{P(x)}$$

CLASIFICADOR NAIVE BAYES

- **Naive ¿ingenuo?**
  - **Suponer que los datos X son independientes de otros valores para la clase C**

Métodos estadísticos: Naive Bayes

EJEMPLO “GOLF”

- **Podremos jugar al golf si:**
  - Cielo = Lluvioso
  - Temperatura= Templado
  - Humedad = Normal
  - Viento = Si
- **Formulación**
  - X={Cielo = Lluvioso; Temperatura= Templado;Humedad = Normal, Viento = Si}
  - **¿Qué es mayor P(X|SI) o P(X|NO) ?**

Cielo	Temperatura	Humedad	Viento	Se jugó
Lluvia	Calor	Alta	No	No
Lluvia	Calor	Alta	Sí	No
Nublado	Calor	Alta	No	Sí
Soleado	Templado	Alta	No	Sí
Soleado	Frío	Normal	No	Sí
Soleado	Frío	Normal	Sí	No
Nublado	Frío	Normal	Sí	Sí
Lluvia	Templado	Alta	No	No
Lluvia	Frío	Normal	No	Sí
Soleado	Templado	Normal	No	Sí
Lluvia	Templado	Normal	Sí	Sí
Nublado	Templado	Alta	Sí	Sí
Nublado	Calor	Normal	No	Sí
Soleado	Templado	Alta	Sí	No

Métodos estadísticos: Naive Bayes

EJEMPLO "GOLF"

$$P(SI) = \frac{9}{14}$$

$$P(NO) = \frac{5}{14}$$

		JUGAR GOLF SI	JUGAR GOLF NO
CIELO	SOLEADO	3 (3/9)	2 (2/5)
	NUBLADO	4 (4/9)	0 (0/5)
	LLUVIOSO	2 (2/9)	3 (3/5)

		JUGAR GOLF SI	JUGAR GOLF NO
TEMPERATURA	CALOR	2 (2/9)	2 (2/5)
	TEMPLADO	4 (4/9)	2 (2 /5)
	FRIO	3 (3/9)	1 (3/5)

		JUGAR GOLF SI	JUGAR GOLF NO
HUMEDAD	ALTA	3 (3/9)	4 (4/5)
	NORMAL	6 (6/9)	1 (1 /5)

		JUGAR GOLF SI	JUGAR GOLF NO
HUMEDAD	ALTA	3 (3/9)	4 (4/5)
	NORMAL	6 (6/9)	1 (1 /5)

Cielo	Temperatura	Humedad	Viento	Se jugó
Lluvia	Calor	Alta	No	No
Lluvia	Calor	Alta	Sí	No
Nublado	Calor	Alta	No	Sí
Soleado	Templado	Alta	No	Sí
Soleado	Frío	Normal	No	Sí
Soleado	Frío	Normal	Sí	No
Nublado	Frío	Normal	Sí	Sí
Lluvia	Templado	Alta	No	No
Lluvia	Frío	Normal	No	Sí
Soleado	Templado	Normal	No	Sí
Lluvia	Templado	Normal	Sí	Sí
Nublado	Templado	Alta	Sí	Sí
Nublado	Calor	Normal	No	Sí
Soleado	Templado	Alta	Sí	No

Métodos estadísticos: Naive Bayes

EJEMPLO "GOLF"

		JUGAR GOLF SI	JUGAR GOLF NO
CIELO	SOLEADO	3 (3/9)	2 (2/5)
	NUBLADO	4 (4/9)	0 (0/5)
	LLUVIOSO	2 (2/9)	3 (3/5)

		JUGAR GOLF SI	JUGAR GOLF NO
TEMPERATURA	CALOR	2 (2/9)	2 (2/5)
	TEMPLADO	4 (4/9)	2 (2 /5)
	FRIO	3 (3/9)	1 (3/5)

		JUGAR GOLF SI	JUGAR GOLF NO
HUMEDAD	ALTA	3 (3/9)	4 (4/5)
	NORMAL	6 (6/9)	1 (1 /5)

		JUGAR GOLF SI	JUGAR GOLF NO
HUMEDAD	ALTA	3 (3/9)	4 (4/5)
	NORMAL	6 (6/9)	1 (1 /5)

X={Cielo = Lluvioso; Temp= Templado; Humedad = Normal, Viento = Si}

C= SI

P(X|SI)= P(Cielo=Lluvia|SI) * P(Temp=Templado|SI) *
P(Humedad=normal|SI) * P(Viento=SI | SI)= 2/9 * 4/9 * 6/9 * 3/9
= 0,219479

P(X|SI) * P(SI) = 0,219479 * 9/14 = **0,0141093**

C=NO

P(X|NO)= P(Cielo=Lluvia|NO) * P(Temp=Templado|NO) *
P(Humedad=normal|NO)*P(Viento=SI|NO)= 3/5 * 2/5 * 1/5 * 3/5
=0,0288

P(X|NO) * P(NO) = 0,0288 * 5/14 = **0,0102857**

Métodos estadísticos: Naive Bayes

EJEMPLO "GOLF"

		JUGAR GOLF SI	JUGAR GOLF NO
CIELO	SOLEADO	3 (3/9)	2 (2/5)
	NUBLADO	4 (4/9)	0 (0/5)
	LLUVIOSO	2 (2/9)	3 (3/5)

		JUGAR GOLF SI	JUGAR GOLF NO
TEMPERATURA	CALOR	2 (2/9)	2 (2/5)
	TEMPLADO	4 (4/9)	2 (2 /5)
	FRIO	3 (3/9)	1 (3/5)

		JUGAR GOLF SI	JUGAR GOLF NO
HUMEDAD	ALTA	3 (3/9)	4 (4/5)
	NORMAL	6 (6/9)	1 (1 /5)

		JUGAR GOLF SI	JUGAR GOLF NO
HUMEDAD	ALTA	3 (3/9)	4 (4/5)
	NORMAL	6 (6/9)	1 (1 /5)

$P(X) = \sum_{i=0}^{n} P(Ci|X)P(Ci)$ = P(SI|X)) * P(SI) * P(NO|X) * P(NO) = **0,0141093**
+ **0,0102857= 0,0243**

P(SI|X) = 0,0141093 / 0,0243 = 0,5783

P(NO|X) = 0,0102857 / 0,0243 =0,4216

Con estas condiciones si que podremos jugar al GOLF

Métodos estadísticos: Naive Bayes

EJEMPLO “CLASIFICADOR DE FLORES”

ATRIBUTO	VALOR
color	amarillo, blanco, rojo
diametro (cm)	5-15,16-25
petalos	5-10, 11-39, 40-70
clase	lirio, margarita, rosa

COLOR	DIAMETRO	PETALOS	CLASE
rojo	5-15	40-70	rosa
blanco	5-15	11-39	margarita
amarillo	5-15	5-10	lirio
blanco	5-15	40-70	rosa
rojo	5-15	40-70	rosa
rojo	5-15	40-70	rosa
blanco	5-15	10-30	margarita
blanco	16-25	5-10	lirio
amarillo	16.25	5-10	lirio
blanco	5-15	40-70	rosa

Calcula el tipo de flor que es una que tiene color rojo un a diámetro entre 16-25 y entre 40-70 pétalos

OJO: Parar al finalizar las matrices de frecuencia

Métodos estadísticos: Naive Bayes

EJEMPLO “CLASIFICADOR DE FLORES”

ATRIBUTO	VALOR
color	amarillo, blanco, rojo
diametro (cm)	5-15,16-25
petalos	5-10, 11-39, 40-70
clase	lirio, margarita, rosa

COLOR	DIAMETRO	PETALOS	CLASE
rojo	5-15	40-70	rosa
blanco	5-15	11-39	margarita
amarillo	5-15	5-10	lirio
blanco	5-15	40-70	rosa
rojo	5-15	40-70	rosa
rojo	5-15	40-70	rosa
blanco	5-15	10-30	margarita
blanco	16-25	5-10	lirio
amarillo	16.25	5-10	lirio
blanco	5-15	40-70	rosa

Total de instancias = 10

P(rosa) = 5/10 = 0,5
P(margarita) = 2/10 = 0,2
P (lirio) = 3/10 = 0,3

Métodos estadísticos: Naive Bayes

EJEMPLO “CLASIFICADOR DE FLORES”

ATRIBUTO	VALOR
color	amarillo, blanco, rojo
diametro (cm)	5-15,16-25
petalos	5-10, 11-39, 40-70
clase	lirio, margarita, rosa

COLOR	DIAMETRO	PETALOS	CLASE
rojo	5-15	40-70	rosa
blanco	5-15	11-39	margarita
amarillo	5-15	5-10	lirio
blanco	5-15	40-70	rosa
rojo	5-15	40-70	rosa
rojo	5-15	40-70	rosa
blanco	5-15	10-30	margarita
blanco	16-25	5-10	lirio
amarillo	16.25	5-10	lirio
blanco	5-15	40-70	rosa

Tablas de frecuencia

COLOR	LIRIO	MARGARITA	ROSA
amarillo	2	0	0
blanco	1	2	2
rojo	0	0	3

DIAMETRO	LIRIO	MARGARITA	ROSA
5-15	1	2	5
16-25	2	0	0

PETALOS	LIRIO	MARGARITA	ROSA
5-10	3	0	0
11-39	0	2	0
40-70	0	0	5

Transformación Lagrange

COLOR	LIRIO	MARGARITA	ROSA
amarillo	3	1	1
blanco	2	3	3
rojo	1	1	4

DIAMETRO	LIRIO	MARGARITA	ROSA
5-15	2	3	6
16-25	3	1	1

PETALOS	LIRIO	MARGARITA	ROSA
5-10	4	1	1
11-39	1	3	1
40-70	1	1	6

Métodos estadísticos: Naive Bayes

EJEMPLO “CLASIFICADOR DE FLORES”

ATRIBUTO	VALOR
color	amarillo, blanco, rojo
diametro (cm)	5-15,16-25
petalos	5-10, 11-39, 40-70
clase	lirio, margarita, rosa

COLOR	DIAMETRO	PETALOS	CLASE
rojo	5-15	40-70	rosa
blanco	5-15	11-39	margarita
amarillo	5-15	5-10	lirio
blanco	5-15	40-70	rosa
rojo	5-15	40-70	rosa
rojo	5-15	40-70	rosa
blanco	5-15	10-30	margarita
blanco	16-25	5-10	lirio
amarillo	16.25	5-10	lirio
blanco	5-15	40-70	rosa

NORMALIZAMOS

COLOR	LIRIO	MARGARITA	ROSA
amarillo	3	1	1
blanco	2	3	3
rojo	1	1	4

COLOR	LIRIO	MARGARITA	ROSA
amarillo	3/6	1/5	1/8
blanco	2/6	3/5	3/8
rojo	1/6	1/5	4/8

DIAMETRO	LIRIO	MARGARITA	ROSA
5-15	2	3	6
16-25	3	1	1
###	1	1	1

DIAMETRO	LIRIO	MARGARITA	ROSA
5-15	2/6	3/5	6/8
16-25	3/6	1/5	1/8
###	1/6	1/5	1/8

PETALOS	LIRIO	MARGARITA	ROSA
5-10	4	1	1
11-39	1	3	1
40-70	1	1	1

PETALOS	LIRIO	MARGARITA	ROSA
5-10	4/6	1/5	1/8
11-39	1/6	3/5	1/8
40-70	1/6	1/5	6/8

Métodos estadísticos: Naive Bayes

EJEMPLO “CLASIFICADOR DE FLORES”

COLOR	LIRIO	MARGARITA	ROSA
amarillo	0,5	0,2	0,125
blanco	0,333	0,6	0,375
rojo	0,166	0,2	0,5

DIAMETRO	LIRIO	MARGARITA	ROSA
5-15	0,333	0,6	0,75
16-25	0,5	0,2	0,125
###	0,166	0,2	0,125

PETALOS	LIRIO	MARGARITA	ROSA
5-10	0,666	0,2	0,125
11-39	0,166	0,6	0,125
40-70	0,166	0,2	0,75

PREGUNTA color=rojo, diámetro=16-25, pétalos=40-50

P(rosa) = 5/10 = 0,5

P(margarita) = 2/10 = 0,2

P (lirio) = 3/10 = 0,3

Métodos estadísticos: Naive Bayes

EJEMPLO “CLASIFICADOR DE FLORES”

COLOR	LIRIO	MARGARITA	ROSA
amarillo	0,5	0,2	0,125
blanco	0,333	0,6	0,375
rojo	0,166	0,2	0,5

DIAMETRO	LIRIO	MARGARITA	ROSA
5-15	0,333	0,6	0,75
16-25	0,5	0,2	0,125
###	0,166	0,2	0,125

PETALOS	LIRIO	MARGARITA	ROSA
5-10	0,666	0,2	0,125
11-39	0,166	0,6	0,125
40-70	0,166	0,2	0,75

PREGUNTA color=rojo, diámetro=16-25, pétalos=40-50

P(rosa) = 5/10 = 0,5
P(margarita) = 2/10 = 0,2
P (lirio) = 3/10 = 0,3

P(r) = P(rojo|rosa) x P(16-25|rosa) x P(40-50|rosa) x P(rosa) =
0,5 x 0,5 x 0,125 x 0,75 x 0,5 = **0,023**

Métodos estadísticos: Naive Bayes

EJEMPLO “CLASIFICADOR DE FLORES”

COLOR	LIRIO	MARGARITA	ROSA
amarillo	0,5	0,2	0,125
blanco	0,333	0,6	0,375
rojo	0,166	0,2	0,5

DIAMETRO	LIRIO	MARGARITA	ROSA
5-15	0,333	0,6	0,75
16-25	0,5	0,2	0,125
###	0,166	0,2	0,125

PETALOS	LIRIO	MARGARITA	ROSA
5-10	0,666	0,2	0,125
11-39	0,166	0,6	0,125
40-70	0,166	0,2	0,75

PREGUNTA color=rojo, diámetro=16-25, pétalos=40-50

P(rosa) = 5/10 = 0,5
P(margarita) = 2/10 = 0,2
P (lirio) = 3/10 = 0,3

P(r) = P(rojo|rosa) x P(16-25|rosa) x P(40-50|rosa) x P(rosa) =
0,5 x 0,5 x 0,125 x 0,75 x 0,5 = **0,023**

P(m) = P(rojo|margarita) x P(16-25|margarita) x P(40-50|margarita) x P(margarita) = 0,2 x 0,2 x 0,2 x 0,2 x 0,2 = **0,016**

Métodos estadísticos: Naive Bayes

EJEMPLO “CLASIFICADOR DE FLORES”

COLOR	LIRIO	MARGARITA	ROSA
amarillo	0,5	0,2	0,125
blanco	0,333	0,6	0,375
rojo	0,166	0,2	0,5

DIAMETRO	LIRIO	MARGARITA	ROSA
5-15	0,333	0,6	0,75
16-25	0,5	0,2	0,125
###	0,166	0,2	0,125

PETALOS	LIRIO	MARGARITA	ROSA
5-10	0,666	0,2	0,125
11-39	0,166	0,6	0,125
40-70	0,166	0,2	0,75

PREGUNTA color=rojo, diámetro=16-25, pétalos=40-50

P(rosa) = 5/10 = 0,5
P(margarita) = 2/10 = 0,2
P (lirio) = 3/10 = 0,3

P(r) = P(rojo|rosa) x P(16-25|rosa) x P(40-50|rosa) x P(rosa) =
0,5 x 0,5 x 0,125 x 0,75 x 0,5 = **0,023**

P(m) = P(rojo|margarita) x P(16-25|margarita) x P(40-50|margarita) x P(margarita) =
0,2 x 0,2 x 0,2 x 0,2 x 0,2 = **0,016**

P(l) = P(rojo|lirio) x P(16-25|lirio) x P(40-50|lirio) x P(lirio) =
0,166 x 0,5 x 0,166 x 0,3 = 0,0041

Métodos estadísticos: Naive Bayes

EJEMPLO "CLASIFICADOR DE FLORES"

COLOR	LIRIO	MARGARITA	ROSA
amarillo	0,5	0,2	0,125
blanco	0,333	0,6	0,375
rojo	0,166	0,2	0,5

DIAMETRO	LIRIO	MARGARITA	ROSA
5-15	0,333	0,6	0,75
16-25	0,5	0,2	0,125
###	0,166	0,2	0,125

PETALOS	LIRIO	MARGARITA	ROSA
5-10	0,666	0,2	0,125
11-39	0,166	0,6	0,125
40-70	0,166	0,2	0,75

PREGUNTA color=rojo, diámetro=16-25, pétalos=40-50

P(rosa) = 5/10 = 0,5
P(margarita) = 2/10 = 0,2
P (lirio) = 3/10 = 0,3

P(r) = P(rojo|rosa) x P(16-25|rosa) x P(40-50|rosa) x P(rosa) =
0,5 x 0,5 x 0,125 x 0,75 x 0,5 = **0,023**

P(m) = P(rojo|margarita) x P(16-25|margarita) x P(40-50|margarita) x P(margarita) =
0,2 x 0,2 x 0,2 x 0,2 x 0,2 = **0,016**

P(l) = P(rojo|lirio) x P(16-25|lirio) x P(40-50|lirio) x P(lirio) =
0,166 x 0,5 x 0,166 x 0,3 = 0,0041

RESPUESTA: Es una rosa

REDES NEURONALES

Métodos estadísticos: Redes Neuronales Artificiales

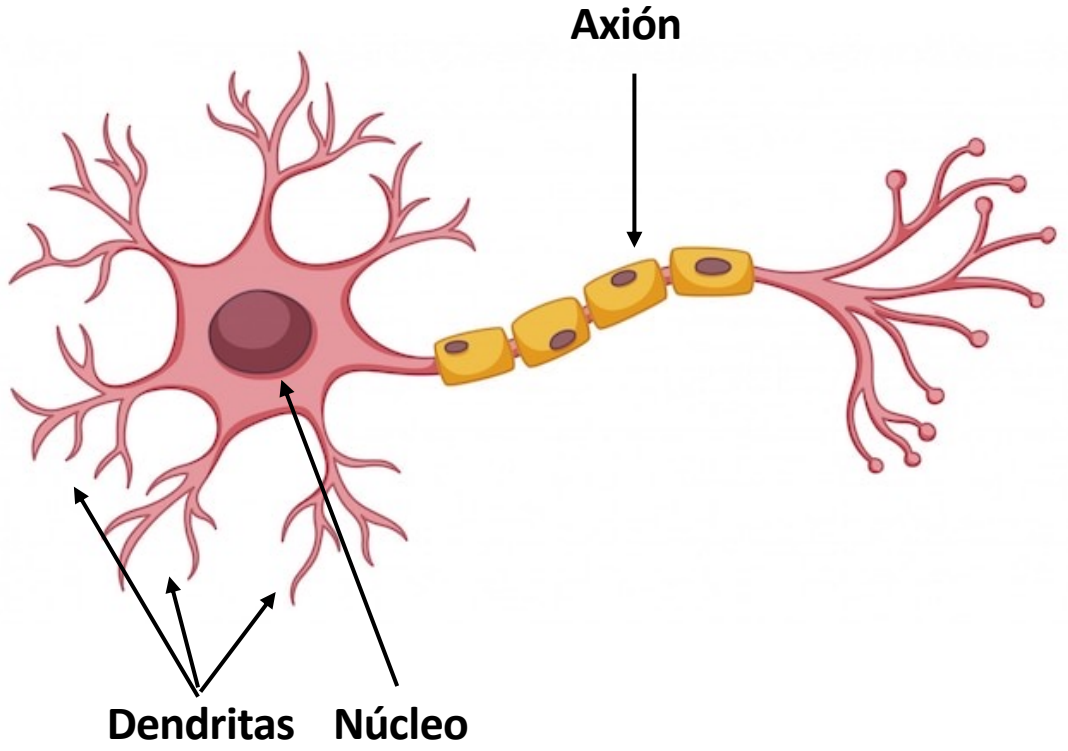
Perceptron Simple

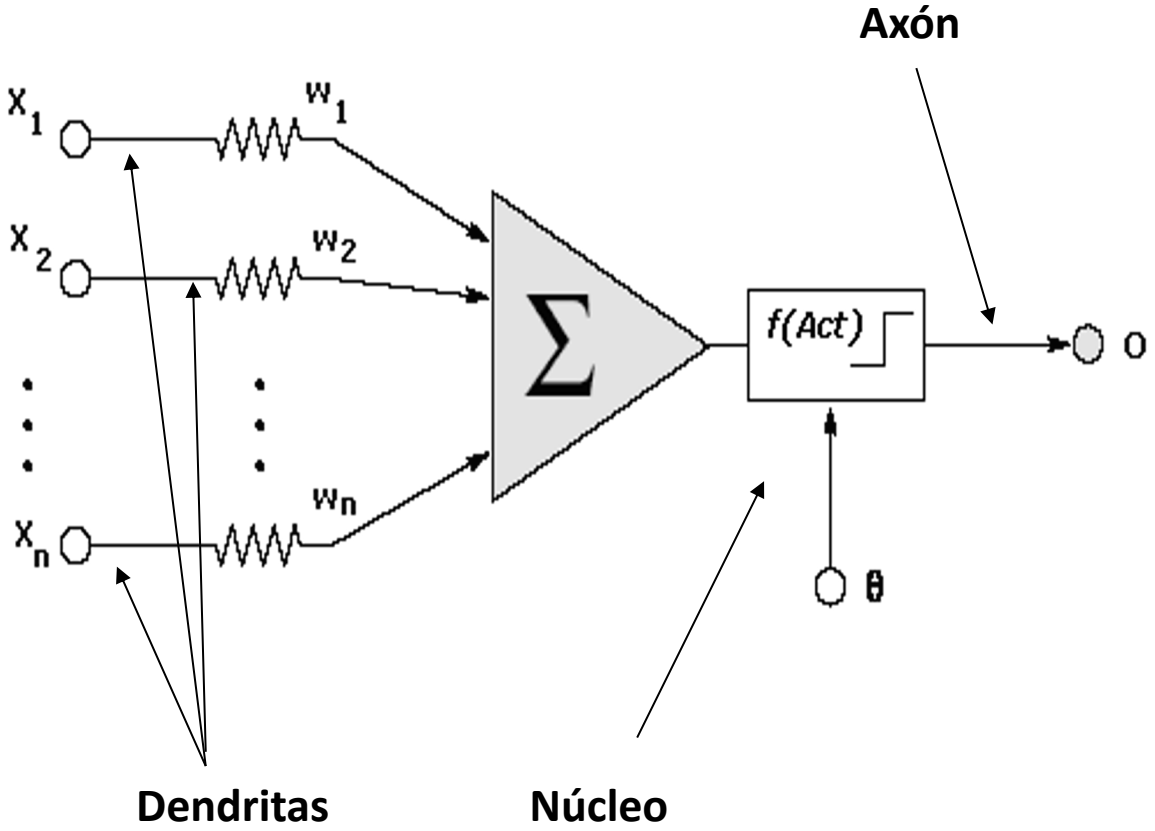
Perceptron Multicapa

PERCEPTRON SIMPLE

Métodos estadísticos: Redes Neuronales Artificiales

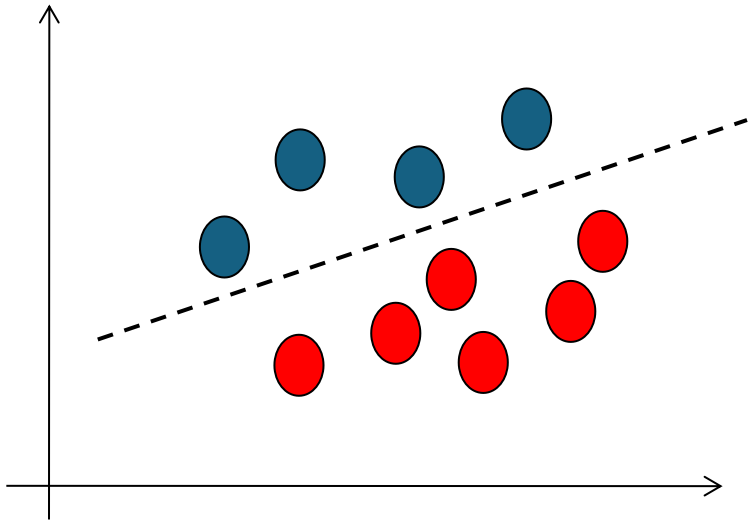
Perceptron Simple





Métodos estadísticos: Perceptron Simple

- Un perpectrón solo es capaz de resolver problemas linealmente separables



- Para sistemas más complejos: Perceptrón multicapa

Métodos estadísticos: Perceptron Simple

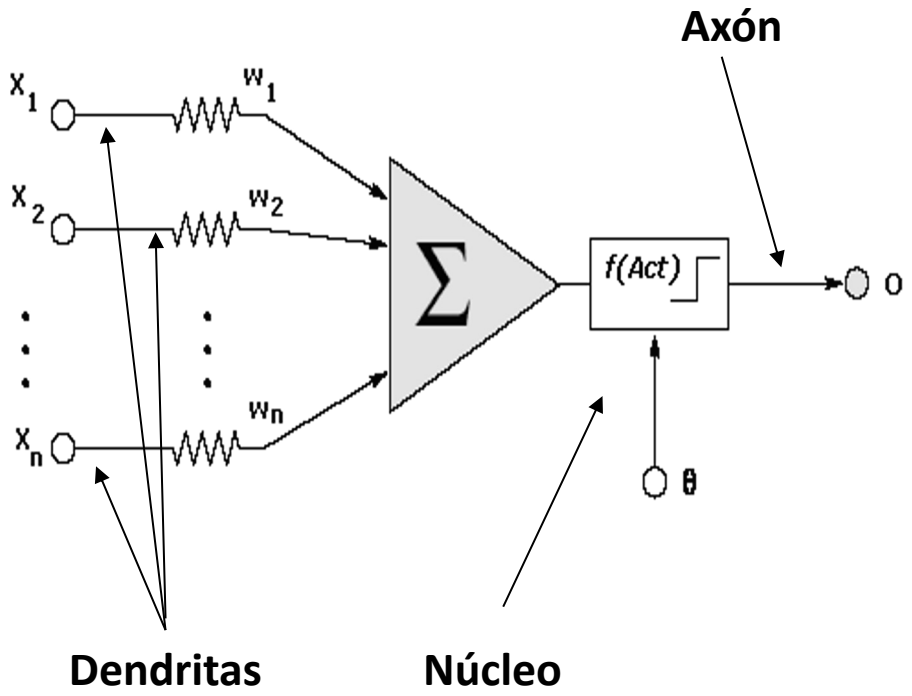
$$act = \sum_{i=1} w_i^j x_i - \theta^j$$

$o = f(act)$ donde a $f(act)$ se le llama función de activación

$$f(Act) = \begin{cases} 1 & si \quad act > 0 \\ -1 & eoc \end{cases}$$

θ^j : Umbral o estado interno de la neurona j-ésima.

w_i^j : peso i-ésimo de la j-ésima neurona.

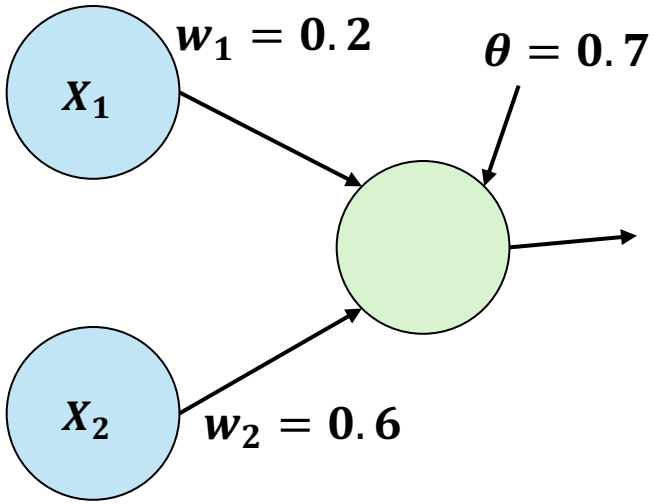


En lo adelante, θ se tratará como un peso adicional en la neurona, el cual pondera el estímulo de una entrada con valor constante igual a -1. Luego:

$$act^j = \sum_{i=1} w_i^j x_i - \theta^j = \sum_{i=0} w_i^j x_i$$

Métodos estadísticos: Perceptron Simple

Función *AND*



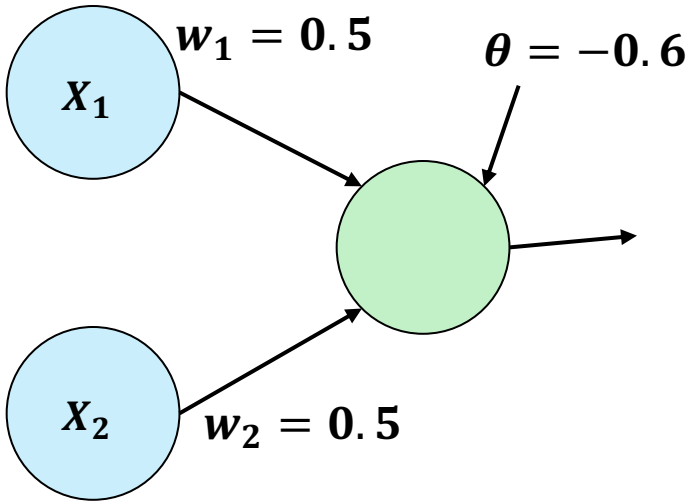
Entrada | Salida esperada | Salida obtenida

X_1	X_2	*AND*	act	$f(act)$
1	1	**1**	0.1	**1**
1	-1	**-1**	-1.1	**-1**
-1	1	**-1**	-0.3	**-1**
-1	-1	**-1**	-1.5	**-1**

$$f(act) = \begin{cases} 1 & si \quad Act > 0 \\ -1 & si \end{cases}$$

Métodos estadísticos: Perceptron Simple

Función *OR*



Entrada		Salida esperada		Salida obtenida
X_1	X_2	*OR*	act	$f(act)$
1	1	**1**	1.6	**1**
1	-1	**1**	0.6	**1**
-1	1	**1**	0.6	**1**
-1	-1	**-1**	-0.4	**-1**

$$f(act) = \begin{cases} 1 & si \quad Act > 0 \\ -1 & eoc \end{cases}$$

Métodos estadísticos: Perceptron Simple

Algoritmo de entrenamiento

Definir función de error: $J^d(\vec{w}) = (t^d - o^d)$

El objetivo es minimizar el error que comete la red!

$$act^j = \sum_{i=0} w_i^j x_i$$

Caso 1: o $= -1$ y $t = 1$

$$f(act) = \begin{cases} 1 & si \quad Act > 0 \\ -1 & eoc \end{cases}$$

!!!Debe aumentarse act¡¡¡

Signo de w_i^j	Signo de x_i^j	Acción
+	+	Aumentar el valor de w_i^j Como ejercicio analice por qué
+	-	La componente $w_i^j x_i^d$ será negativa, por lo tanto hay que disminuir (acercar a 0 por la derecha) el valor del peso para que esta influya menos en act^{td} .
-	+	La componente $w_i^j x_i^d$ será negativa, por lo tanto hay que aumentar (acercar a 0 por la izquierda) el valor del peso para que esta influya menos en act^{td}.
-	-	Disminuir (alejar de 0 por la izquierda) el valor de w_i^j. Como ejercicio analice por qué.

Métodos estadísticos: Perceptron Simple

Algoritmo de entrenamiento

Definir función de error: $J^d(\vec{w}) = (t^d - o^d)$

El objetivo es minimizar el error que comete la red!

Caso 2: o = 1 y $t = -1$

!!!Debe disminuirse act¡¡¡

$$act^j = \sum_{i=0} w_i^j x_i$$

$$f(act) = \begin{cases} 1 & si \quad Act > 0 \\ -1 & eoc \end{cases}$$

Signo de w_i^j	Signo de x_i	Acción
+	+	Disminuir el valor de w_i^j
+	-	La componente $w_i^j x_i^d$ será negativa, por lo tanto hay que aumentar el valor del peso para hacerla más negativa aún.
-	+	La componente $w_i^j x_i^d$ será negativa, por lo tanto hay que disminuir el valor del peso para hacerla más negativa aún
-	-	Aumentar (acercar a 0 por la izquierda) el valor de w_i^j

Métodos estadísticos: Perceptron Simple

Algoritmo de entrenamiento

Comienzo:

1- Inicializar los w_{ij} (peso correspondiente a la entrada i-ésima del j-ésimo perceptrón)

2- Seleccionar un vector $\langle x^d, t^d \rangle \in CE$ y calcular o^{jd}

3- Cambiar los pesos de acuerdo a:
$w_{ij} = w_{ij} + \Delta w_{ij}$ donde $\Delta w_{ij} = \eta(t^{jd} - o^{jd})x_i^d$

4- Volver al paso 2.

Fin

CE: Conjunto de entrenamiento

η: Factor de aprendizaje

t^{jd}: Salida esperada para el j-ésimo perceptrón

o^{jd}: Salida dada por el j-ésimo perceptrón

Métodos estadísticos: Perceptron Simple

Algoritmo de entrenamiento

Comienzo:

1- Inicializar los w_{ij} (peso correspondiente a la entrada i-ésima del j-ésimo perceptrón)

2- Seleccionar un vector $\langle x^d, t^d \rangle \in CE$ y calcular o^{jd}

3- Cambiar los pesos de acuerdo a:
$w_{ij} = w_{ij} + \Delta w_{ij}$ donde $\Delta w_{ij} = \eta(t^{jd} - o^{jd})x_i^d$

4- Volver al paso 2.

Fin

CE: Conjunto de entrenamiento

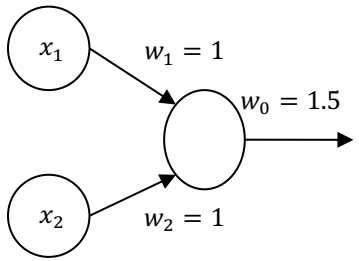
η: Factor de aprendizaje

t^{jd}: Salida esperada para el j-ésimo perceptrón

o^{jd}: Salida dada por el j-ésimo perceptrón

Ejercicio

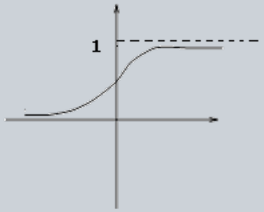
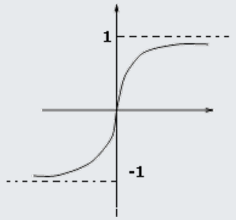
Empleando el algoritmo de entrenamiento del perceptrón ajuste los pesos de la red representada a continuación de modo que aprenda la función NAND.



X_1	X_2	$NAND$
1	1	**-1**
1	-1	**1**
-1	1	**1**
-1	-1	**1**

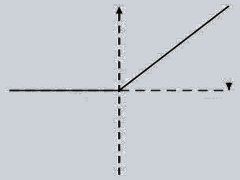
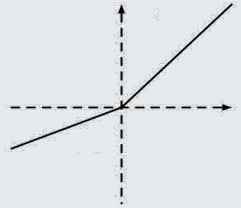
Métodos estadísticos: Perceptron Simple

Funciones de activación

Funciones de Activación	Expresión	Gráfica	Derivada
Sigmoide logística	$f(act^j) = \sigma(act^j) = \frac{1}{1+e^{-act^j}}$		$f'(Act^j) = \sigma(act^j)(1-\sigma(act^j))$
Tangente Hiperbólica	$f(Act^j) = tanh(act^j)$ $= \frac{sinh(act^j)}{cosh(act^j)}$ $= \frac{e^{-act^j} - e^{-act^j}}{e^{act^j} + e^{-act^j}}$		$f'(Act^j) = 1 - tanh(act^j)$

Métodos estadísticos: Perceptron Simple

Funciones de activación

Funciones de Activación	Expresión	Gráfica	Derivada
ReLU (Rectified Linear Unit)	$f(act^j) = max(0, act^j)$		$f'(Act^j) = \begin{cases} 0 & si \quad act^j < 0 \\ 1 & si \quad act^j > 0 \\ nan & si \quad act^j = 0 \end{cases}$
Leaky ReLU	$f(act^j) = max(\alpha\ act^j, act^j)$ $\alpha \in [0,1]$		$f'(Act^j) = \begin{cases} \alpha & si \quad act^j < 0 \\ 1 & si \quad act^j \geq 0 \end{cases}$

Métodos estadísticos: Perceptron Simple

Funciones de activación

Funciones de Activación	Expresión	Derivada
Softmax	$f(\overrightarrow{act}) = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix} = \begin{bmatrix} \frac{e^{act_1}}{\sum_i e^{act_i}} \\ \frac{e^{act_2}}{\sum_i e^{act_i}} \\ \vdots \\ \frac{e^{act_n}}{\sum_i e^{act_i}} \end{bmatrix}$	$f'(\overrightarrow{act}) = \begin{bmatrix} \frac{\partial p_1}{\partial act_1} & \frac{\partial p_1}{\partial act_2} & \cdots & \frac{\partial p_1}{\partial act_n} \\ \frac{\partial p_2}{\partial act_1} & \frac{\partial p_2}{\partial act_2} & \cdots & \frac{\partial p_2}{\partial act_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial p_n}{\partial act_1} & \frac{\partial p_n}{\partial act_2} & \cdots & \frac{\partial p_n}{\partial act_n} \end{bmatrix}$ $\frac{\partial p_i}{\partial act_j} = \begin{cases} p_i(1-p_j) & si \quad i = j \\ -p_i p_j & si \quad i \neq j \end{cases}$

Métodos estadísticos: Perceptron Simple

Regla Deltha o de Widrow-Hoff. Variante on-line

Regla Deltha Variante On-line

Comienzo:

1 - Inicializar w_i^j aleatoriamente.

2 - Hasta que la condición de parada se satisfaga:

2.1 - Seleccionar un vector $<\vec{x}^d, \vec{t}^d> \in CE$ y calcular $\vec{o}^{jd}$

.

2.2 - Cambiar los pesos de acuerdo a:

$w_i^j = w_i^j + \Delta w_i^j$ donde $\Delta w_i^j = \eta(t^{jd} - o^{jd}) f'(Act^{jd}) x_i^d$

2.3 - Volver al paso **2**.

Fin.

Regla Deltha o de Widrow-Hoff. Variante batch

Regla Deltha Variante Batch

Comienzo:

1 - Inicializar w_i^j aleatoriamente. $\Delta w_i^j = 0$

2 - Hasta que la condición de parada se satisfaga:

2.1 - Para cada vector $<\vec{x}^d, \vec{t}^d> \in CE$ calcular:

- $\vec{o}^{jd}$
- $\Delta w_i^{jd} = (t^{jd} - o^{jd}) f'(Act^{jd}) x_i^d$
- $\Delta w_i^j = \Delta w_i^j + \Delta w_i^{jd}$

2.2 - Cambiar los pesos de acuerdo con:

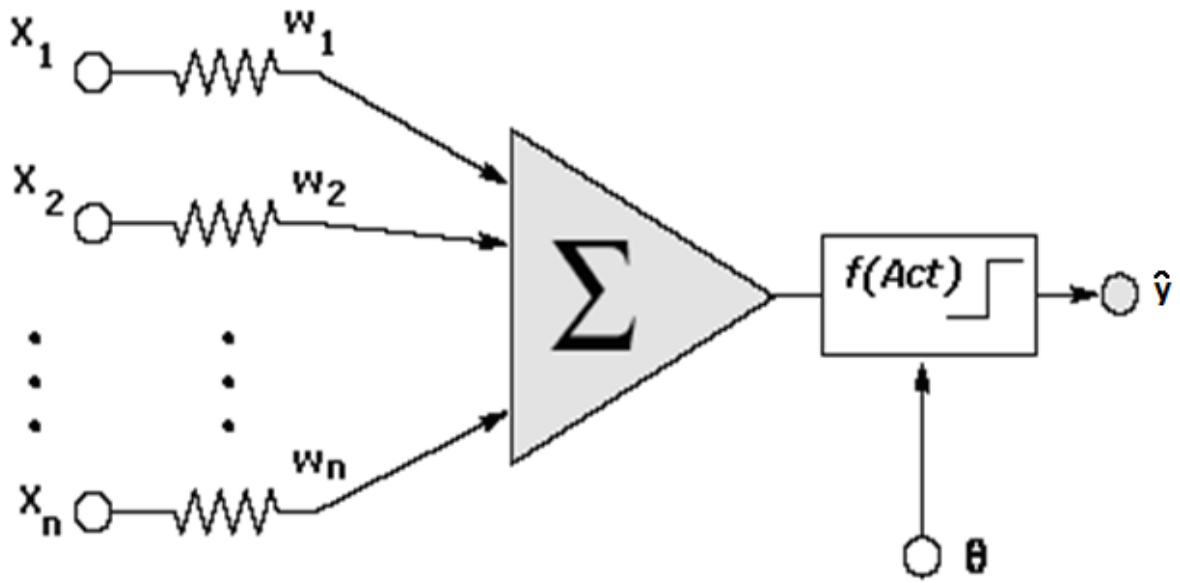
$w_i^j = w_i^j + \eta \Delta w_i^j$

2.3 - Volver al paso **2**.

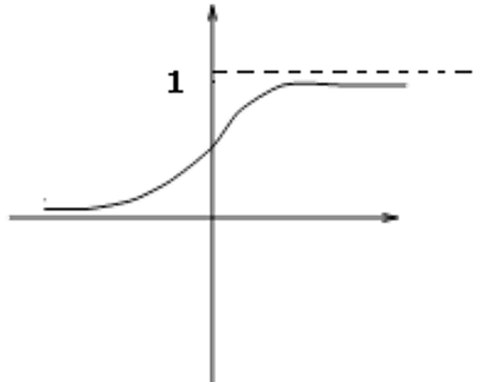
Fin

Métodos estadísticos: Perceptron Simple

Regresión logística



$$Act = \sum_i w_i x_i - \theta \qquad \hat{y} = f(Act) = \frac{1}{1 + e^{-\alpha Act}}$$



Métodos estadísticos: Perceptron Simple

Regresión logística

- La Regresión Logística se emplea en aprendizaje supervisado, cuando la salida y es 0 o 1.

- Notar que $0 \leq \hat{y} \leq 1$ por lo que se interpreta como $\hat{y} = P(y = 1|x)$

- Se puede utilizar para clasificación binaria de dos categorías A y B, haciendo $\hat{y} = P(y = A|x)$, clasificando como A si $\hat{y} > T$, donde $T = 0.5$ en general.

$$Act = \sum_i w_i x_i - \theta$$

$$\hat{y} = f(Act) = \frac{1}{1 + e^{-\alpha Act}}$$

Métodos estadísticos: Perceptron Simple

Regresión logística

Loss function

Mide la discrepancia entre la predicción $\hat{y}$ y la salida deseada y

$$L(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$$

en la práctica se emplea

$$L(\hat{y}, y) = -(y \log \hat{y} + (1 - y) \log(1 - \hat{y}))$$

$$Act = \sum_i w_i x_i - \theta$$

$$\hat{y} = f(Act) = \frac{1}{1 + e^{-\alpha Act}}$$

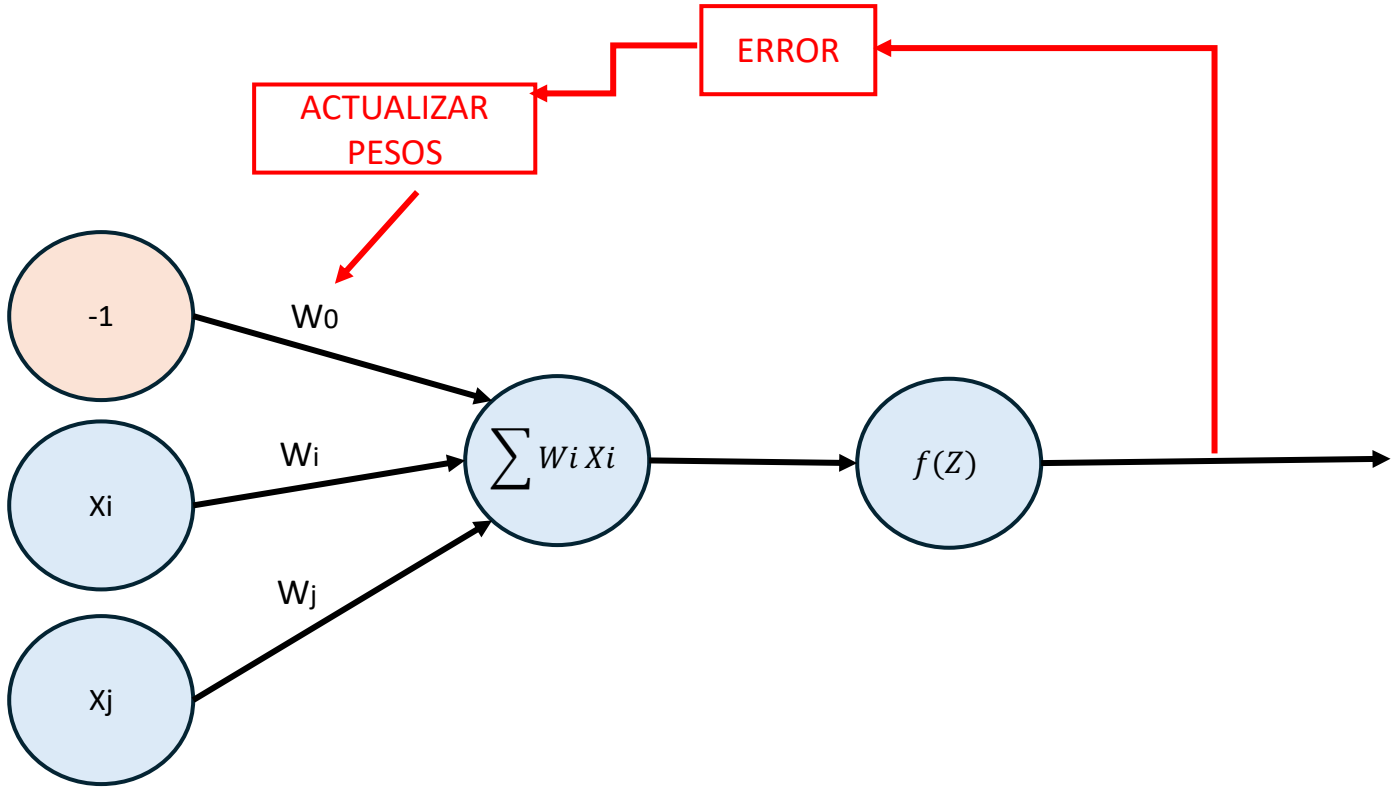
Cost function

Caracteriza el promedio de la función de pérdida en todo el conjunto de entrenamiento

$$J(w, \theta) = \frac{1}{m} \sum_{j=1}^{m} L\left(\hat{y}^{(j)}, y^{(j)}\right) = -\frac{1}{m} \sum_{j=1}^{m} \left(y^{(j)} \log \hat{y}^{(j)} + (1 - y^{(j)}) \log(1 - \hat{y}^{(j)})\right)$$

- **EJEMPLO APRENDIZAJE**

MUESTRA	MATRIZ X		f(z). v. esperado
1	X1	1	1
	X2	1	
2	X1	1	1
	X2	-1	
3	X1	-1	1
	X2	1	
4	X1	-1	-1
	X2	-1	



- **EJEMPLO APRENDIZAJE**

MUESTRA	MATRIZ X		f(z). v. esperado
1	X0	-1	1
	X1	1	
	X2	1	
2	X0	-1	1
	X1	1	
	X2	-1	
3	X0	-1	1
	X1	-1	
	X2	1	
4	X0	-1	-1
	X1	-1	
	X2	-1	

X0= Bias

$\eta = 0{,}5$

$Z = \sum_{i=0}^{2} Xi * Wi$

$$F(Z)\begin{cases} 1 \ si \ z \geq 0 \\ -1 \ si \ z < 0 \end{cases}$$

ITERACIÓN 1

1

		Xi*Wi	Z	f(Z)	Esperado	ERROR= δ		
X0	-1	0	0	1	1	0	$\Delta 0=\mu*\delta*X0$	0
W0	**0**						$\Delta 1=\mu*\delta*X1$	0
X1	1	0					$\Delta 2=\mu*\delta*X2$	0
W1	**0**						$W0=W0+\Delta 0$	0
X2	1	0					$W1=W1+\Delta 1$	0
W2	**0**						$W2=W2+\Delta 2$	0

2

		Xi*Wi	Z	f(Z)	Esperado	ERROR= δ		
X0	-1	0	0	1	1	0	$\Delta 0=\mu*\delta*X0$	0
W0	**0**						$\Delta 1=\mu*\delta*X1$	0
X1	1	0					$\Delta 2=\mu*\delta*X2$	0
W1	**0**						$W0=W0+\Delta 0$	0
X2	-1	0					$W1=W1+\Delta 1$	0
W2	**0**						$W2=W2+\Delta 2$	0

- **EJEMPLO APRENDIZAJE**

MUESTRA	MATRIZ X		f(z). v. esperado
1	X0	-1	1
	X1	1	
	X2	1	
2	X0	-1	1
	X1	1	
	X2	-1	
3	X0	-1	1
	X1	-1	
	X2	1	
4	X0	-1	-1
	X1	-1	
	X2	-1	

X0= Bias

$\eta = 0{,}5$

Z= $\sum_{i=0}^{2} Xi * Wi$

F(Z) $\begin{cases} 1 \ si \ z \geq 0 \\ -1 \ si \ z < 0 \end{cases}$

ITERACIÓN 1

3

		Xi*Wi	Z	f(Z)	Esperado	ERROR= δ		
X0	-1	0	0	1	1	0	$\Delta 0=\mu*\delta*X0$	0
W0	0						$\Delta 1=\mu*\delta*X1$	0
X1	-1	0					$\Delta 2=\mu*\delta*X2$	0
W1	0						$W0=W0+\Delta 0$	0
X2	1	0					$W1=W1+\Delta 1$	0
W2	0						$W2=W2+\Delta 2$	0

4

		Xi*Wi	Z	f(Z)	Esperado	ERROR= δ		
X0	-1	0	0	1	-1	-2	$\Delta 0=\mu*\delta*X0$	1
W0	0						$\Delta 1=\mu*\delta*X1$	1
X1	-1	0					$\Delta 2=\mu*\delta*X2$	1
W1	0						$W0=W0+\Delta 0$	1
X2	-1	0					$W1=W1+\Delta 1$	1
W2	0						$W2=W2+\Delta 2$	1

- **EJEMPLO APRENDIZAJE**

MUESTRA	MATRIZ X		f(z). v. esperado
1	X0	-1	1
	X1	1	
	X2	1	
2	X0	-1	1
	X1	1	
	X2	-1	
3	X0	-1	1
	X1	-1	
	X2	1	
4	X0	-1	-1
	X1	-1	
	X2	-1	

X0= Bias

$\eta = 0{,}5$

Z= $\sum_{i=0}^{2} Xi * Wi$

F(Z) $\begin{cases} 1 \ si \ z \geq 0 \\ -1 \ si \ z < 0 \end{cases}$

ITERACIÓN 2

1

		Xi*Wi	Z	f(Z)	Esperado	ERROR= δ	$\Delta i=\mu*\delta*Xi$	
X0	-1	-1	1	1	1	0	$\Delta 0=\mu*\delta*X0$	0
W0	**1**						$\Delta 1=\mu*\delta*X1$	0
X1	1	1					$\Delta 2=\mu*\delta*X2$	0
W1	**1**						$W0=W0+\Delta 0$	1
X2	1	1					$W1=W1+\Delta 1$	1
W2	**1**						$W2=W2+\Delta 2$	1

2

		Xi*Wi	Z	f(Z)	Esperado	ERROR= δ		
X0	-1	-1	-1	-1	1	2	$\Delta 0=\mu*\delta*X0$	-1
W0	**1**						$\Delta 1=\mu*\delta*X1$	1
X1	1	1					$\Delta 2=\mu*\delta*X2$	-1
W1	**1**						$W0=W0+\Delta 0$	0
X2	-1	-1					$W1=W1+\Delta 1$	2
W2	**1**						$W2=W2+\Delta 2$	0

- **EJEMPLO APRENDIZAJE**

MUESTRA	MATRIZ X		f(z). v. esperado
1	X0	-1	1
	X1	1	
	X2	1	
2	X0	-1	1
	X1	1	
	X2	-1	
3	X0	-1	1
	X1	-1	
	X2	1	
4	X0	-1	-1
	X1	-1	
	X2	-1	

X0= Bias

$\eta = 0{,}5$

$Z = \sum_{i=0}^{2} Xi * Wi$

$$F(Z)\begin{cases} 1 \ si \ z \geq 0 \\ -1 \ si \ z < 0 \end{cases}$$

ITERACIÓN 2

3								
		Xi*Wi	Z	f(Z)	Esperado	ERROR= δ		
X0	-1	0	-2	-1	1	2	$\Delta 0=\mu*\delta*X0$	-1
W0	**0**						$\Delta 1=\mu*\delta*X1$	-1
X1	-1	-2					$\Delta 2=\mu*\delta*X2$	1
W1	**2**						$W0=W0+\Delta 0$	-1
X2	1	0					$W1=W1+\Delta 1$	1
W2	**0**						$W2=W2+\Delta 2$	1

4								
		Xi*Wi	Z	f(Z)	Esperado	ERROR= δ		
X0	-1	1	-1	-1	-1	0	$\Delta 0=\mu*\delta*X0$	0
W0	**-1**						$\Delta 1=\mu*\delta*X1$	0
X1	-1	-1					$\Delta 2=\mu*\delta*X2$	0
W1	**1**						$W0=W0+\Delta 0$	-1
X2	-1	-1					$W1=W1+\Delta 1$	1
W2	**1**						$W2=W2+\Delta 2$	1

- **EJEMPLO APRENDIZAJE**

MUESTRA	MATRIZ X		f(z). v. esperado
1	X0	-1	1
	X1	1	
	X2	1	
2	X0	-1	1
	X1	1	
	X2	-1	
3	X0	-1	1
	X1	-1	
	X2	1	
4	X0	-1	-1
	X1	-1	
	X2	-1	

X0= Bias

$\eta = 0{,}5$

$Z = \sum_{i=0}^{2} Xi * Wi$

$$F(Z)\begin{cases} 1 \ si \ z \geq 0 \\ -1 \ si \ z < 0 \end{cases}$$

ITERACIÓN 3

1		Xi*Wi	Z	f(Z)	Esperado	ERROR= δ	$\Delta i=\mu*\delta*Xi$	
X0	-1	1	3	1	1	0	$\Delta 0=\mu*\delta*X0$	0
W0	**-1**						$\Delta 1=\mu*\delta*X1$	0
X1	1	1					$\Delta 2=\mu*\delta*X2$	0
W1	**1**						$W0=W0+\Delta 0$	-1
X2	1	1					$W1=W1+\Delta 1$	1
W2	**1**						$W2=W2+\Delta 2$	1

2		Xi*Wi	Z	f(Z)	Esperado	ERROR= δ		
X0	-1	1	1	1	1	0	$\Delta 0=\mu*\delta*X0$	0
W0	**-1**						$\Delta 1=\mu*\delta*X1$	0
X1	1	1					$\Delta 2=\mu*\delta*X2$	0
W1	**1**						$W0=W0+\Delta 0$	-1
X2	-1	-1					$W1=W1+\Delta 1$	1
W2	**1**						$W2=W2+\Delta 2$	1

- **EJEMPLO APRENDIZAJE**

MUESTRA	MATRIZ X		f(z). v. esperado
1	X0	-1	1
	X1	1	
	X2	1	
2	X0	-1	1
	X1	1	
	X2	-1	
3	X0	-1	1
	X1	-1	
	X2	1	
4	X0	-1	-1
	X1	-1	
	X2	-1	

X0= Bias

$\eta = 0{,}5$

$Z = \sum_{i=0}^{2} Xi * Wi$

$$F(Z)\begin{cases} 1 \ si \ z \geq 0 \\ -1 \ si \ z < 0 \end{cases}$$

ITERACIÓN 3

3		Xi*Wi	Z	f(Z)	Esperado	ERROR= δ		
X0	-1	1	1	1	1	0	*Δ0=μ*δ*X0*	0
W0	**-1**						*Δ1=μ*δ*X1*	0
X1	-1	-1					*Δ2=μ*δ*X2*	0
W1	**1**						*W0=W0+Δ0*	-1
X2	1	1					*W1=W1+Δ1*	1
W2	**1**						*W2=W2+Δ2*	1

4		Xi*Wi	Z	f(Z)	Esperado	ERROR= δ		
X0	-1	1	-1	-1	-1	0	*Δ0=μ*δ*X0*	0
W0	**-1**						*Δ1=μ*δ*X1*	0
X1	-1	-1					*Δ2=μ*δ*X2*	0
W1	**1**						*W0=W0+Δ0*	-1
X2	-1	-1					*W1=W1+Δ1*	1
W2	**1**						*W2=W2+Δ2*	1

- **EJEMPLO APRENDIZAJE**

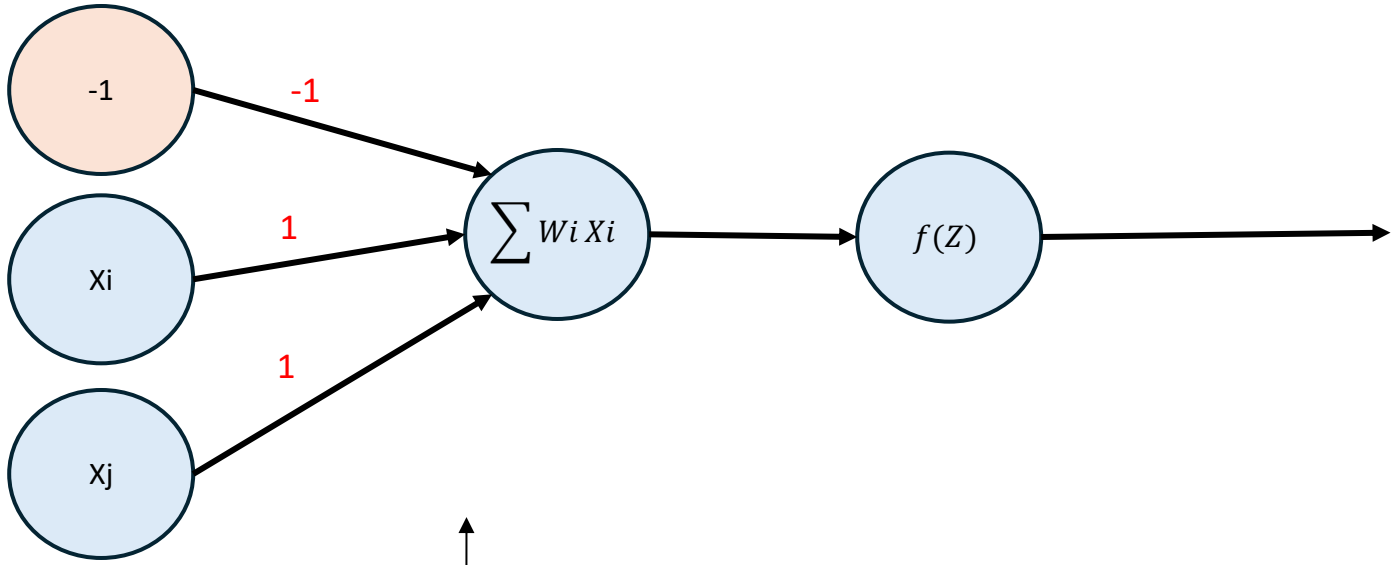
MUESTRA	MATRIZ X		f(z). v. esperado
1	X1	1	1
	X2	1	
2	X1	1	1
	X2	-1	
3	X1	-1	1
	X2	1	
4	X1	-1	-1
	X2	-1	

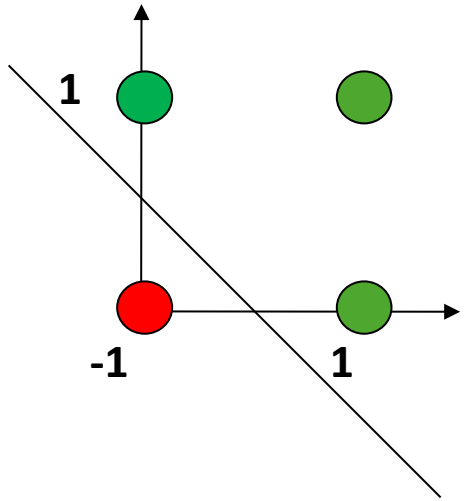
X0= Bias

$\eta = 0{,}5$

Z= $\sum_{i=0}^{2} Xi * Wi$

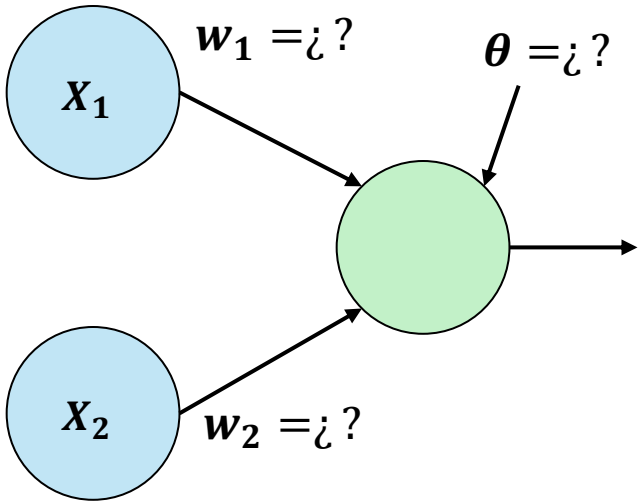
F(Z) $\begin{cases} 1 \ si \ z \geq 0 \\ -1 \ si \ z < 0 \end{cases}$





Métodos estadísticos: Perceptron Simple

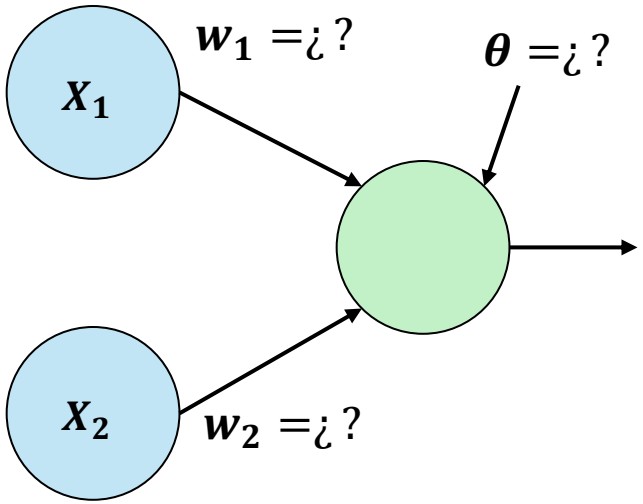
¿Podrá una neurona lineal representar la función XOR?



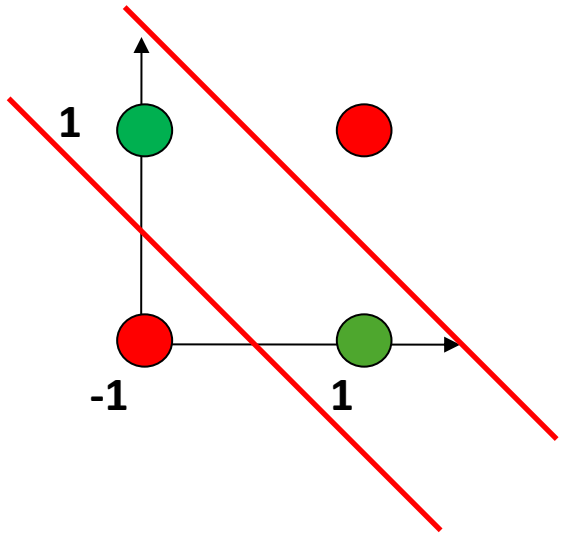
X_1	X_2	XOR	act	$f(act)$
1	1	**-1**		
1	-1	**1**		
-1	1	**1**		
-1	-1	**-1**		

Métodos estadísticos: Perceptron Simple

¿Podrá una neurona lineal representar la función XOR?



X_1	X_2	XOR	act	$f(act)$
1	1	**-1**		
1	-1	**1**		
-1	1	**1**		
-1	-1	**-1**		

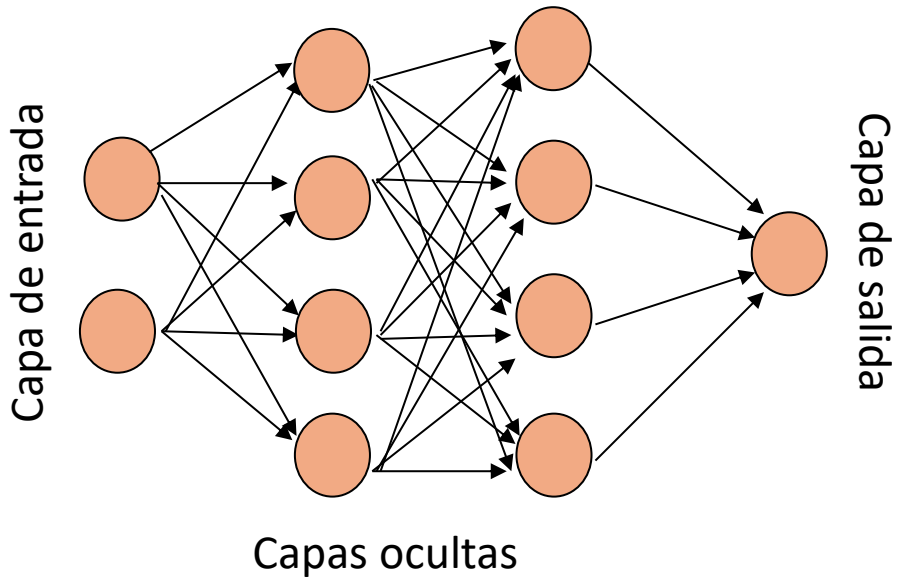


PERCEPTRON MULTICAPA

Métodos estadísticos: Perceptron Multicapa

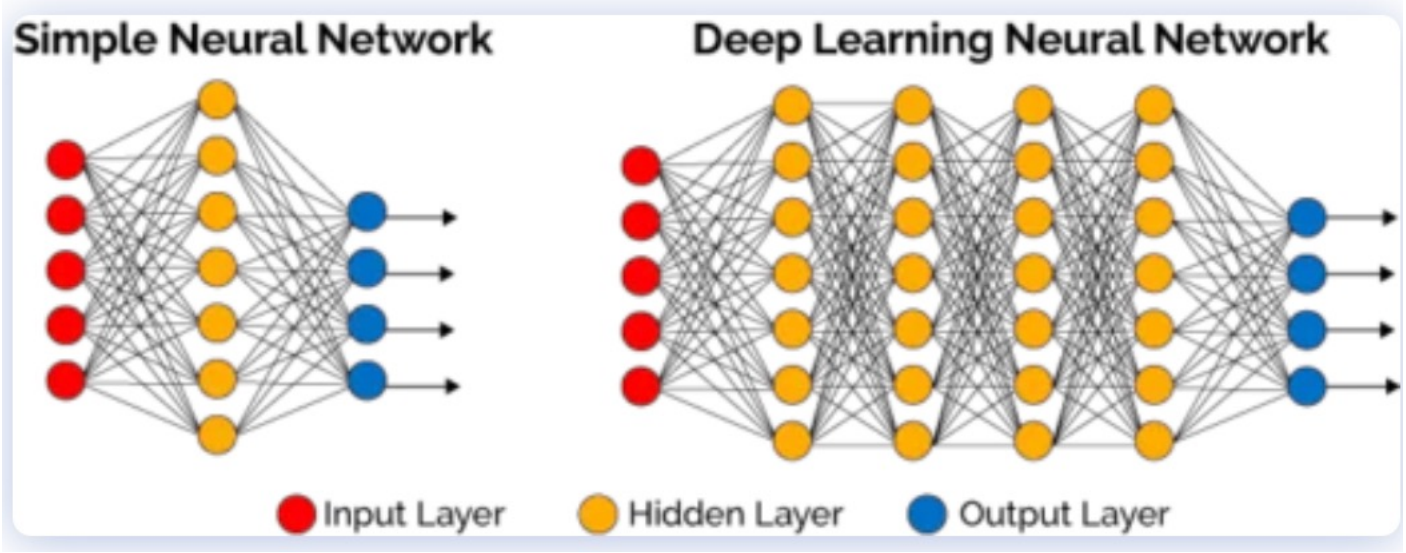
Red *feedforward*

- Entrenamiento:
  - Generalización regla delta
  - Algoritmo *backpropagation*



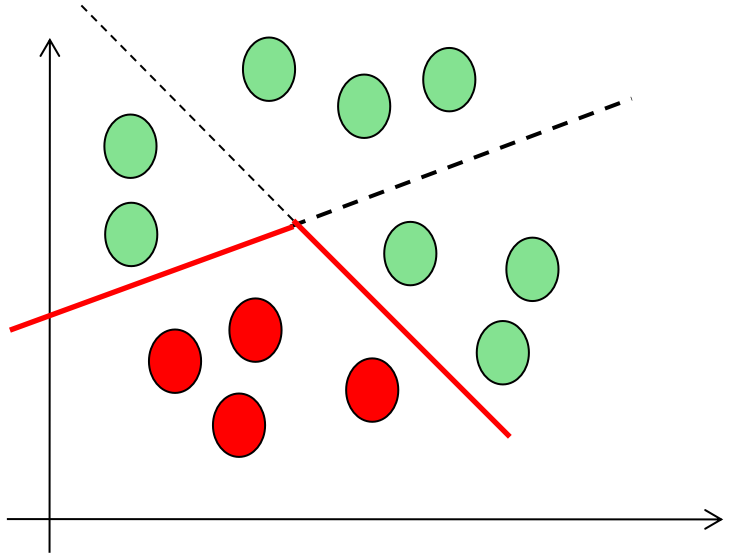
Métodos estadísticos: Perceptron Multicapa

- Ejemplos de sistemas no linealmente separables que se pueden resolver

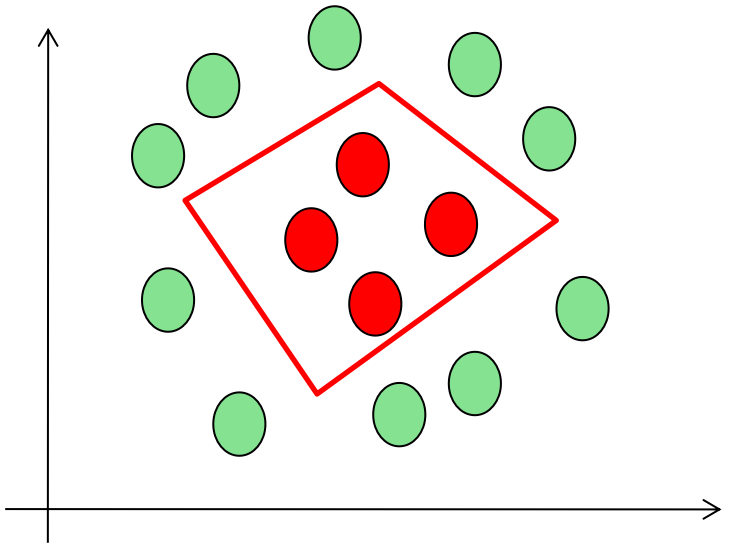


Métodos estadísticos: Perceptron Multicapa

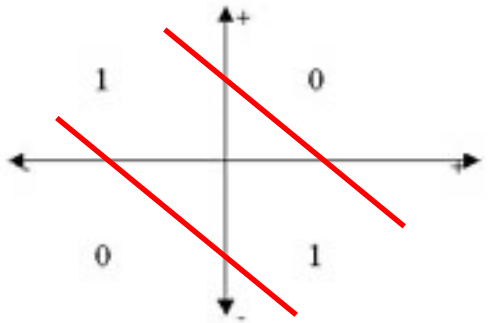
- Ejemplos de sistemas no linealmente separables que se pueden resolver



EJEMPLO 1



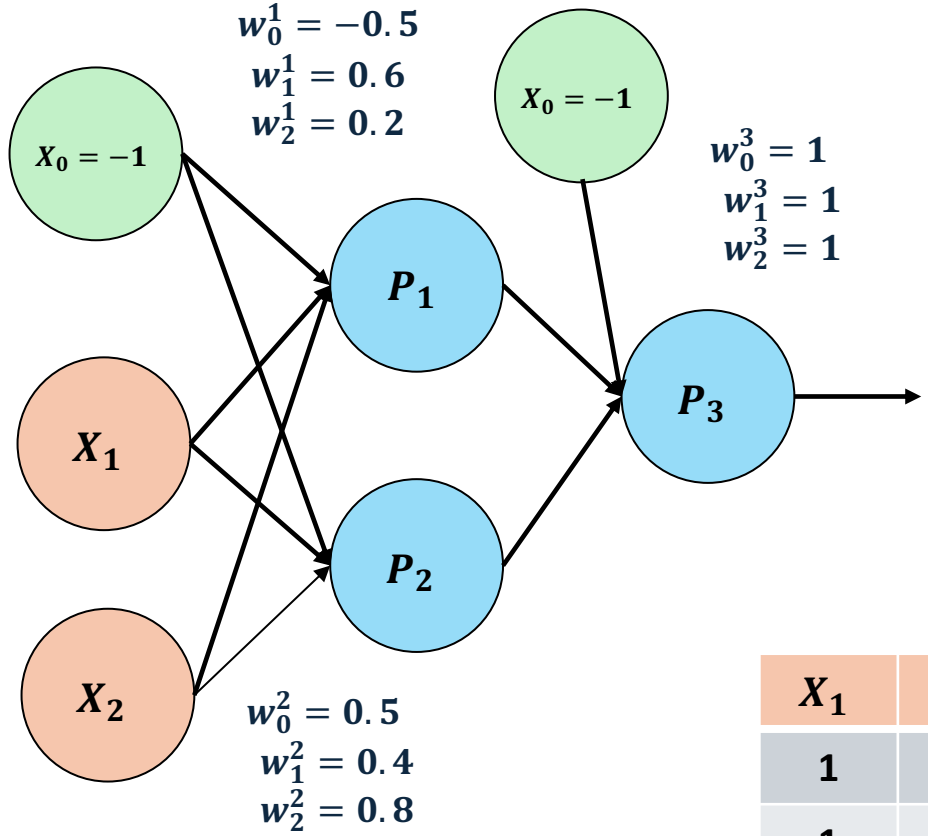
EJEMPLO 2



EJEMPLO XOR

Métodos estadísticos: Perceptron Multicapa

Función XOR

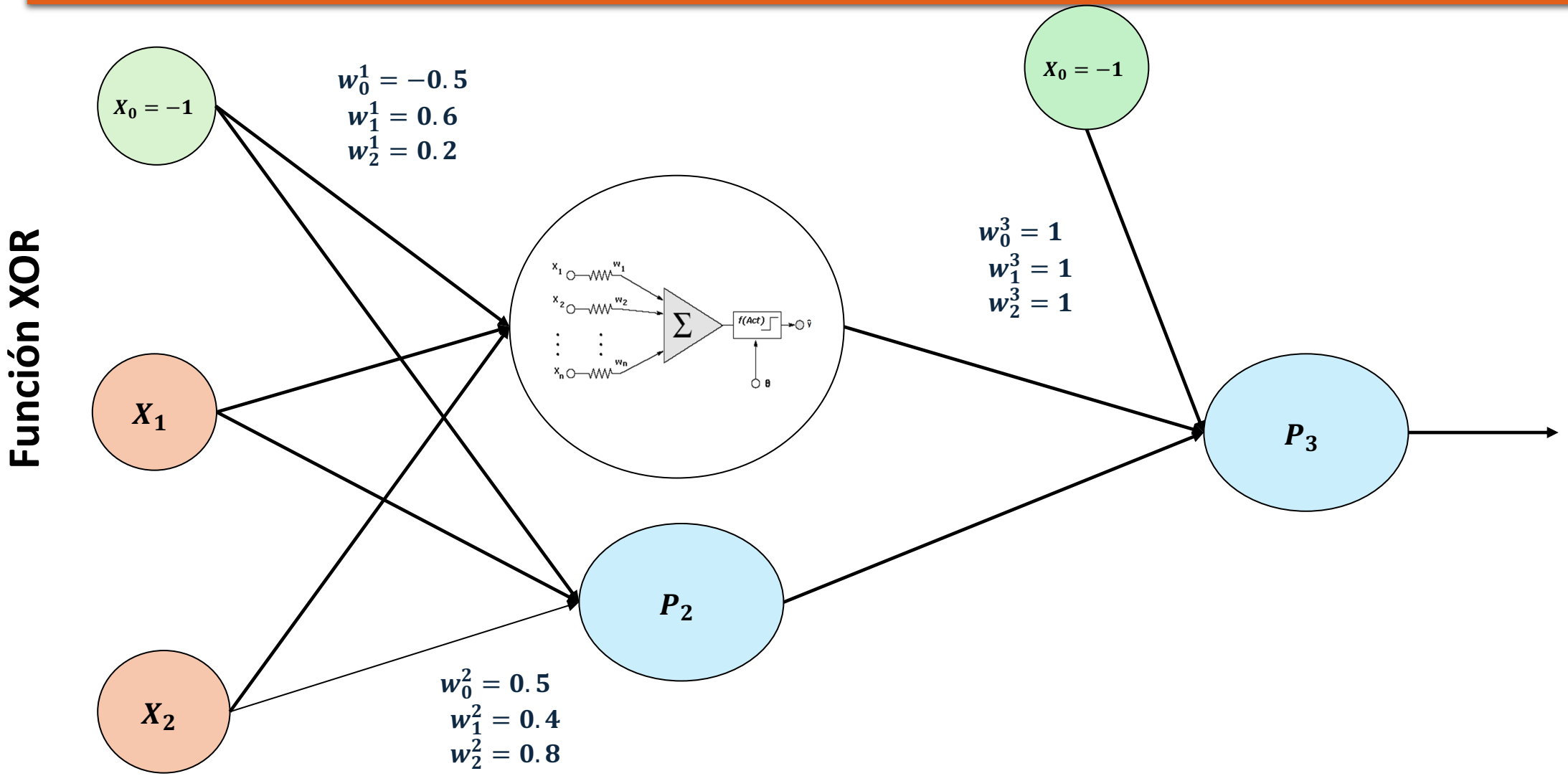


X_1	X_2	XOR
1	1	-1
1	-1	1
-1	1	1
-1	-1	-1

X_1	X_2	P_1	P_2
1	1	1	1
1	-1	1	-1
-1	1	1	-1
-1	-1	-1	-1

P_1	P_2	P_3
1	1	-1
1	-1	1
1	-1	1
-1	-1	-1

Métodos estadísticos: Perceptron Multicapa



Métodos estadísticos: Perceptron Multicapa

Backpropagation Variante On-line (estocástica)

Comienzo:

1 - Inicializar w_j^i aleatoriamente. $\Delta w_j^i = 0$

2 - Hasta que la condición de parada se satisfaga:

2.1 - Para cada vector $<\vec{x}^d, \vec{t}^d> \in CE$ calcular:

- $\vec{o}^d$
- Para las neuronas de la capa de salida: $\delta^o = (t^{od} - o^{od}) f'(Act^{od})$
- Para las neuronas de las capas ocultas: $\delta^{hd} = \sum_{j=1}^{C} \delta^{jd} w_h^j f'(Act^{hd})$
- Cambiar los pesos de acuerdo con: $w_i^j = w_i^j + \eta \Delta w_i^{dj}$ donde $\Delta w_i^{dj} = \eta \delta^{jd} o^{id}$

2.2 - Volver al paso **2**.

Fin.

C : Cantidad de neuronas en la capa siguiente a la neurona h.

w_h^j : Peso con que la neurona j pondera el estímulo recibido desde la neurona h.

Notar que o^{id} es la salida de la i-ésima neurona, siendo a su vez un impulso transmitido a las neuronas de la siguiente capa.

Backpropagation Variante Batch

Comienzo:

1 - Inicializar w_j^i aleatoriamente. $\Delta w_j^i = 0$

2 - Hasta que la condición de parada se satisfaga:

2.1 - Para cada vector $<\vec{x}^d, \vec{t}^d> \in CE$ calcular:

- $\vec{o}^d$
- Para las neuronas de la capa de salida: $\delta^{od} = (t^{od} - o^{od}) f'(Act^{od})$
- Para las neuronas de las capas ocultas: $\delta^{hd} = \sum_{j=1}^{C} \delta^{jd} w_h^j f'(Act^{hd})$
- Hacer $\Delta w_i^j = \Delta w_i^j + \eta \delta^{jd} o^{id}$

2.2 Cambiar los pesos de acuerdo con:

$w_i^j = w_i^j + \Delta w_i^{dj}$

2.3 - Volver al paso **2**.

Fin.

C : Cantidad de neuronas en la capa siguiente a la neurona h.

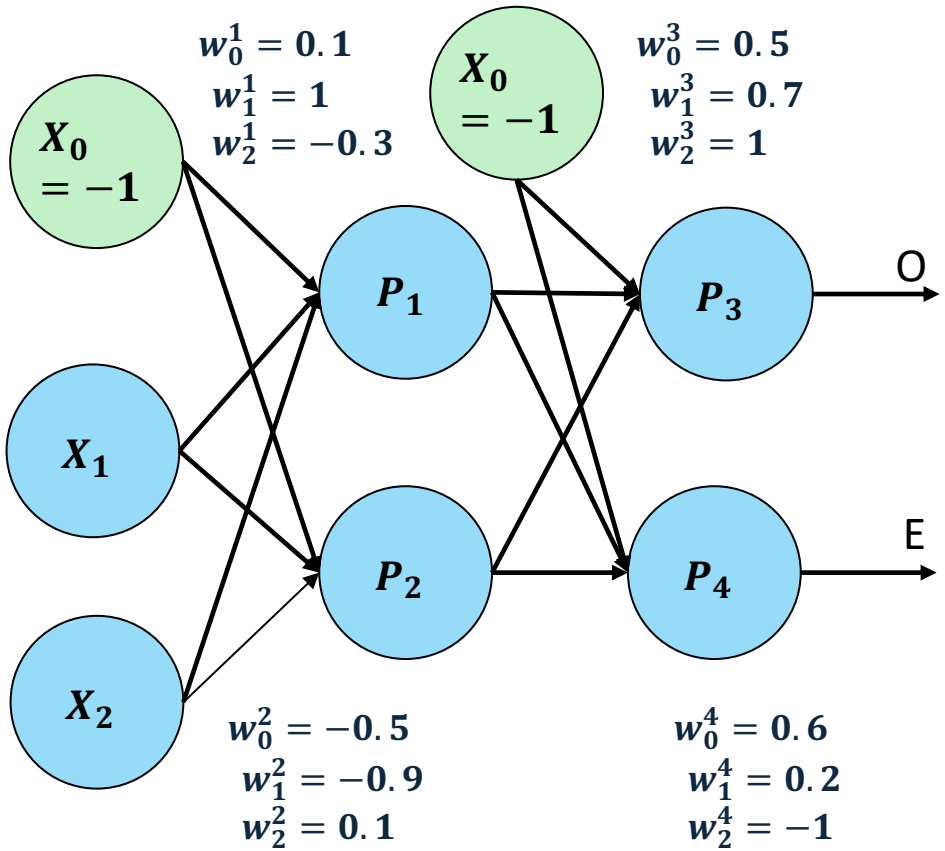
w_h^j : Peso con que la neurona j pondera el estímulo recibido desde la neurona h.

Notar que o^{id} es la salida de la i-ésima neurona, siendo a su vez un impulso transmitido a las neuronas de la siguiente capa.

- Explicación con el ejemplo del café del entrenamiento perceptrón multicapa
  - https://www.youtube.com/watch?v=eNIqz_noix8 (min 5:47)

Métodos estadísticos: Perceptron Multicapa

Ejemplo. Variante On-line



X_1	X_2	O-E
1	1	1,-1
1	-1	1,1
-1	1	1,1
-1	-1	1,-1

- Tomando como función de activación a la **función identidad y(X) = X** y factor aprendizaje $\eta = 0,2$
- Tomar un ejemplo del conjunto de entrenamiento, por ejemplo el primero.
- Calcular la salida de la red, para esto primero calcular la respuesta de P_1 y P_2

$$Act^{P1} = w_0^{P1}x_0 + w_1^{P1}x_1 + w_2^{P1}x_2 = (0.1*-1) + (1*1) + (-0.3*1) = 0.6$$

$$Act^{P2} = w_0^{P2}x_0 + w_1^{P2}x_1 + w_2^{P2}x_2 = (-0.5*-1) + (-0.9*1) + (0.1*1) = -0.3$$

$$Act^{P3} = w_0^{P3}x_0 + w_1^{P3}o^{P1} + w_2^{P3}o^{P2} = (0.5*-1) + (0.7*0.6) + (1*-0.3) = -0.4$$

$$Act^{P4} = w_0^{P4}x_0 + w_1^{P4}o^{P1} + w_2^{P4}o^{P2} = (0.6*-1) + (0.2*0.6) + (-1*-0.3) = -0.2$$

Métodos estadísticos: Perceptron Multicapa

Ejemplo. Variante On-line

Calcular. δ^{P3} y δ^{P4} $\quad f(x) = x \quad f'(x) = 1$

$$\delta^{P3} = (t^{P3} - o^{P3}) f'(Act^{P3}) = (-1-(-0.4))*1 = -0.6$$

$$\delta^{P4} = (t^{P4} - o^{P4}) f'(Act^{P4}) = (1-(-0.2))*1 = 1.2$$

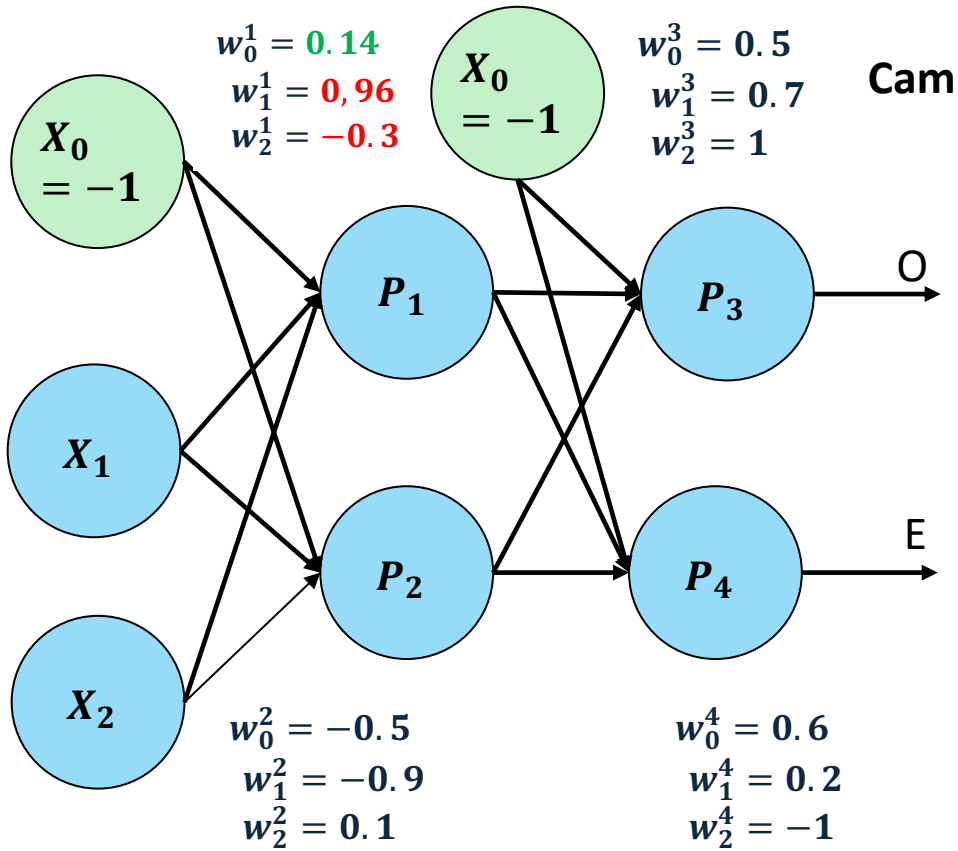
Calcular δ^{P1} y δ^{P2}

$$\delta^{P1} = (\delta^{P3} w_{P1}^{P3} + \delta^{P4} w_{P1}^{P4}) * f'(Act^{P1}) = ((-0.6*0.7)+(1.2*0.2))*1 = -0.2$$

$$\delta^{P2} = (\delta^{P3} w_{P2}^{P3} + \delta^{P4} w_{P2}^{P4}) * f'(Act^{P2}) = ((-0.6*1)+(1.2*-1)*1 = -1.8$$

Métodos estadísticos: Perceptron Multicapa

Ejemplo. Variante On-line



Cambiar los pesos

$$w_0^{P1} = w_0^{P1} + \eta \Delta w_0^{P1} = w_0^{P1} + \eta \delta^{P1} x_0 = 0.1 + 0.2 * (-0.2) * (-1) = 0.14$$

$$w_1^{P1} = w_1^{P1} + \eta \Delta w_1^{P1} = w_1^{P1} + \eta \delta^{P1} x_1 = 1 + 0.2 * (-0.2) * (1) = 0.96$$

$$w_2^{P1} = w_2^{P1} + \eta \Delta w_2^{P1} = w_2^{P1} + \eta \delta^{P1} x_2 = -0.3 + 0.2 * (-0.2) * (1) = -0.34$$

Ejercicio: Calcular la actualización de los restantes pesos en la red.

¿Qué mecanismo escoger?

Criterios de selección del modelo

- Dos decisiones fundamentales:
  - El tipo de modelo (árboles de decisión, redes neuronales, modelos probabilísticos, etc.)
  - El algoritmo utilizado para construir o ajustar el modelo a partir de las instancias de entrenamiento (existen varias maneras de construir árboles de decisión, varias maneras de construir redes neuronales, etc.)

Selección del modelo y/o algoritmo

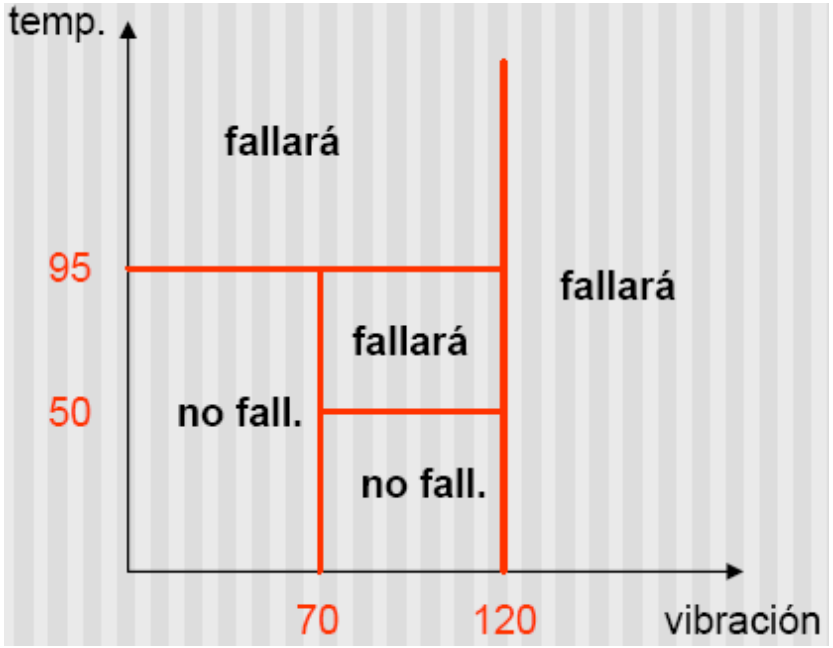
1. Capacidad de representación.
2. Legibilidad.
3. Tiempo de cómputo on-line.
4. Tiempo de cómputo off-line.
5. Dificultad de ajuste de parámetros.
6. Robustez ante el ruido.
7. Sobreajuste.
8. Minimización del error.

Selección del modelo y/o algoritmo
1. Capacidad de representación

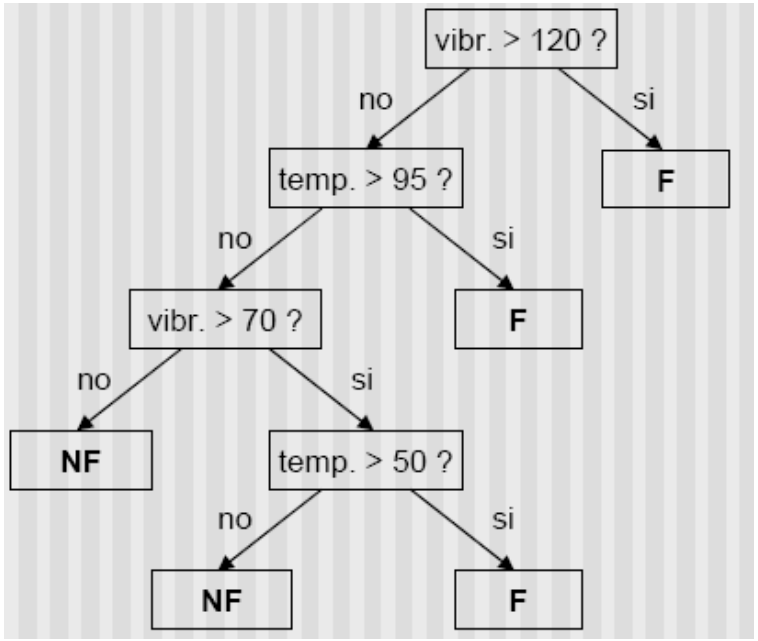
- Relacionado con el tipo de fronteras de decisión que se pueden expresar.
- Fronteras de decisión: separación de clases distintas.
- Cada modelo crea diferentes fronteras.

Selección del modelo y/o algoritmo. Capacidad de representación

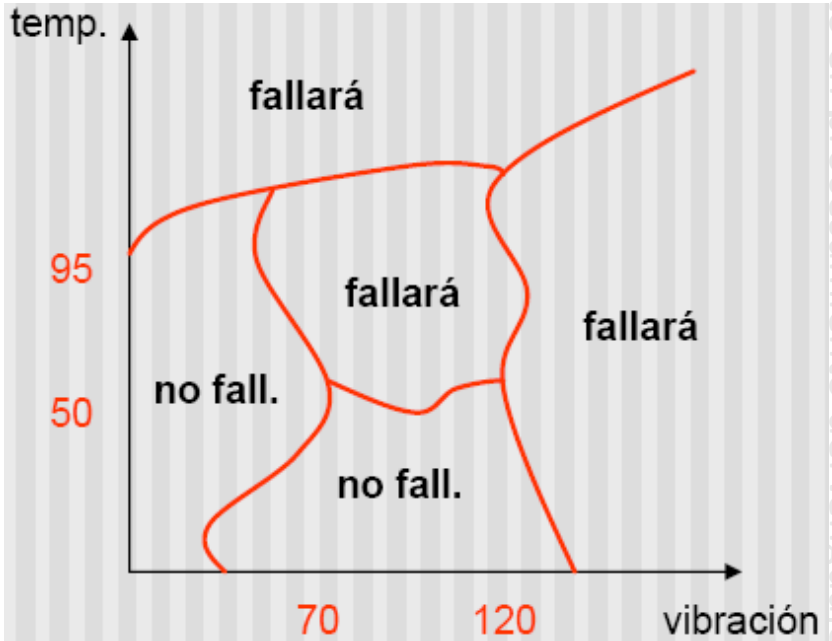
Ejemplo con sólo dos atributos:



Árboles de decisión: fronteras perpendiculares a los ejes



Redes Neuronales (NN), fronteras no lineales:



- Mayor capacidad de representación.
- Permiten representar conceptos más complejos que los árboles de decisión

Selección del modelo y/o algoritmo.
2. Legibilidad

- Capacidad de ser leído e interpretado por un humano.
- Árboles de decisión: fáciles de entender e interpretar:
  - conjunto de reglas.
  - en los niveles más altos están los atributos más importantes.
- Redes neuronales: difíciles (o imposibles) de interpretar: - pesos de conexiones entre neuronas.
- Un modelo legible puede ofrecer información sobre el problema que se estudia (ej. indicar qué atributos afectan la probabilidad de fallo de una máquina y cómo).
- Un modelo no legible sólo puede ser usado como un clasificador (ej. permite predecir si una máquina fallará o no aplicando el modelo).

Selección del modelo y/o algoritmo
3. Tiempo de cómputo on-line

- Es el tiempo necesario para clasificar una instancia:
  - Árboles de decisión: tiempo necesario para recorrer el árbol, evaluando las funciones lógicas de cada nodo.
  - Redes neuronales: tiempo necesario para realizar las operaciones (sumas, productos, sigmoides) incluidas en la red.
- Este tiempo se consume cada vez que se debe clasificar una nueva instancia.
- Algunas aplicaciones requieren clasificar miles de instancias.
  - Ejemplo: clasificación de cada uno de los píxeles de una imagen aérea de un cultivo, río, ruta, etc.
  - Se requiere clasificar millones de instancias.
  - El tiempo de cómputo es muy importante.

Selección del modelo y/o algoritmo
4. Tiempo de cómputo off-line

- Es el tiempo necesario para construir o ajustar el modelo a partir de los ejemplos de entrenamiento.
  - Árboles de decisión: tiempo necesario para elegir la estructura del árbol, los atributos a situar en cada nodo y la optimización mediante la poda.
  - Redes neuronales: tiempo necesario para ajustar los pesos de las conexiones (puede tomar valores muy grandes).
- Sólo se consume una vez, cuando mediante la utilización de los ejemplos de entrenamiento se genera y selecciona el resultado (modelo o hipótesis) más adecuado.
- Dependiendo de la aplicación no es un problema que el tiempo de cómputo off-line sea elevado (se deja una computadora procesando uno o tres días enteros).

Selección del modelo y/o algoritmo
5. Dificultad de ajuste de parámetros

- Se prefieren los algoritmos con pocos (o ninguno) parámetros que ajustar.
- Se prefieren algoritmos con muy poca sensibilidad a la modificación de sus parámetros.
- Hay modelos muy difíciles de ajustar mediante parámetros (puede ocurrir con redes neuronales).

Selección del modelo y/o algoritmo
6. Robustez ante el ruido

- Instancia de entrenamiento ruidosa:
  - etiquetada incorrectamente (ejemplo: una máquina que no falló, etiquetada como que sí falló).
  - algún atributo no está valorizado.
- Algunos algoritmos pueden funcionar adecuadamente aunque haya instancias ruidosas en el conjunto de entrenamiento (ej. árboles de decisión, redes neuronales).
- Otros algoritmos no ofrecen buenos resultados (ej. k-vecinos más cercanos).

Selección del modelo y/o algoritmo
7. Sobreajuste (*overfitting*).

- Problema muy común.
- El modelo está demasiado ajustado a las instancias y no funciona adecuadamente con nuevos casos.
- El modelo no es capaz de generalizar.
- Normalmente, fronteras de decisión muy complejas producen sobreajuste.

Selección del modelo y/o algoritmo
7. Sobreajuste (*overfitting*).

- Ejemplo con dos atributos: