

ALGORITMO ID3



Árbol de decisión

Un **árbol de decisión** es un modelo de aprendizaje automático y toma de decisiones que representa un flujo lógico estructurado en forma de árbol, donde cada nodo implica una pregunta o condición, cada rama una posible respuesta o consecuencia, y cada hoja (nodo final) una decisión o resultado. Es ampliamente usado en **clasificación y regresión**.

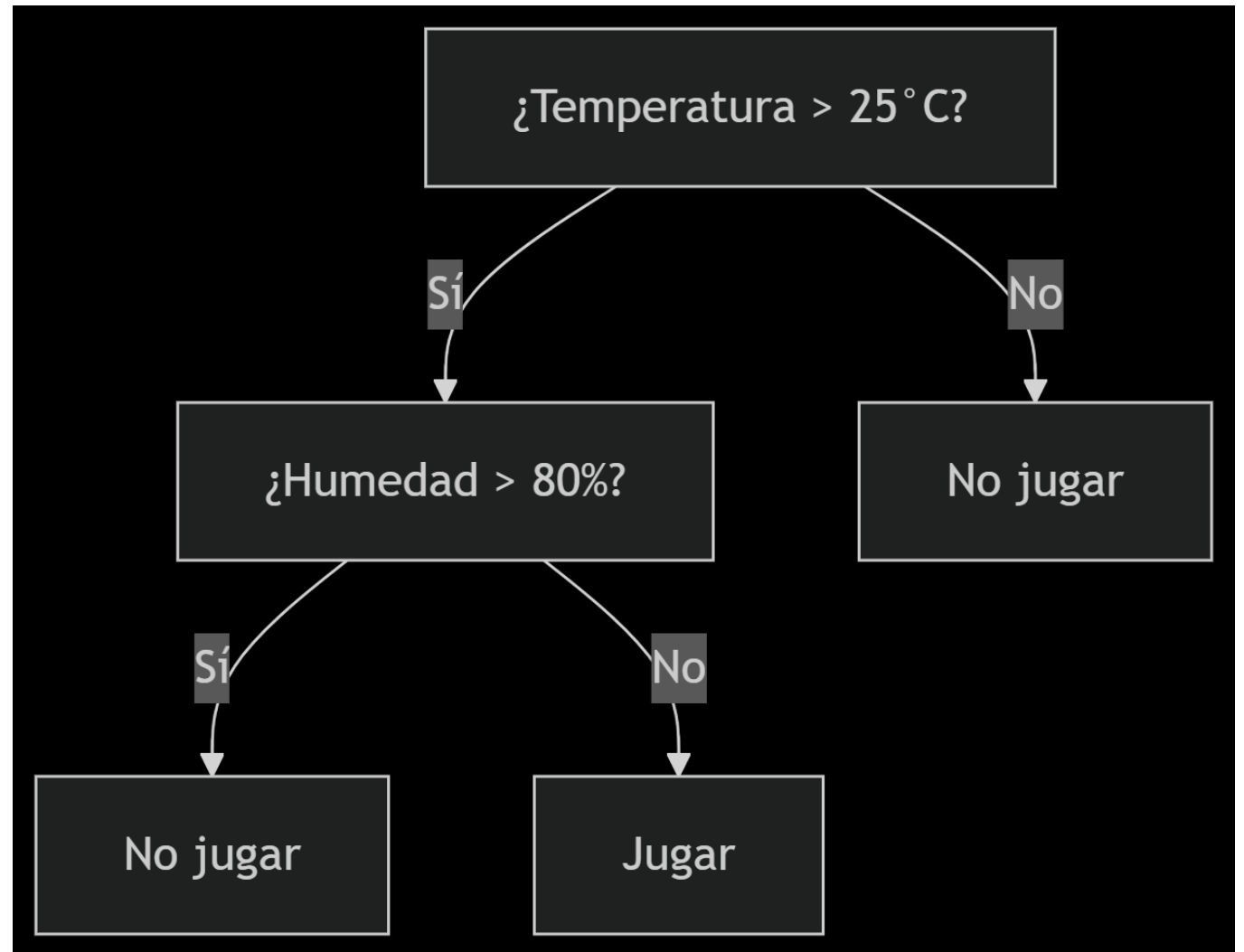
Partes de un árbol de decisión

1. Nodo raíz: La primera pregunta/condición sobre los datos (ej: "¿Es la temperatura > 25°C?").

2. Nodos internos: Preguntas intermedias que dividen los datos (ej: "¿Humedad > 80%?").

3. Ramas: Resultados de cada condición (ej: "Sí" o "No").

4. Hojas (nodos terminales): La decisión final (ej: "Jugar al tenis: Sí").



Como se construye

Se usan algoritmos (como **ID3**, **C4.5**, o **CART**) que usan métricas para elegir las mejores divisiones:

- **Ganancia de información** (basada en entropía).
- **Índice Gini** (mide la impureza de los nodos).

Construcción paso a paso:

- Seleccionar el atributo que mejor divide los datos (maximizando la ganancia).
- Repetir el proceso en cada subconjunto hasta cumplir un criterio (ej: profundidad máxima).

ID3

- El algoritmo ID₃ (Iterative Dichotomiser 3 [dicotomizado iterativo]) es un algoritmo de aprendizaje automático utilizado para construir árboles de decisión, desarrollado por Ross Quinlan.
- Este algoritmo se basa en la estrategia "divide y vencerás" para clasificar objetos o situaciones, tomando como entrada un conjunto de ejemplos caracterizados por un conjunto de atributos, donde uno de ellos es el objetivo a clasificar, generalmente de tipo binario (sí/no, positivo/negativo).
- El proceso de construcción del árbol es recursivo, comenzando desde un nodo raíz y dividiendo los datos en subconjuntos según los valores de los atributos, con el objetivo de crear ramas cada vez más homogéneas en cuanto a la clase objetivo

Resumen

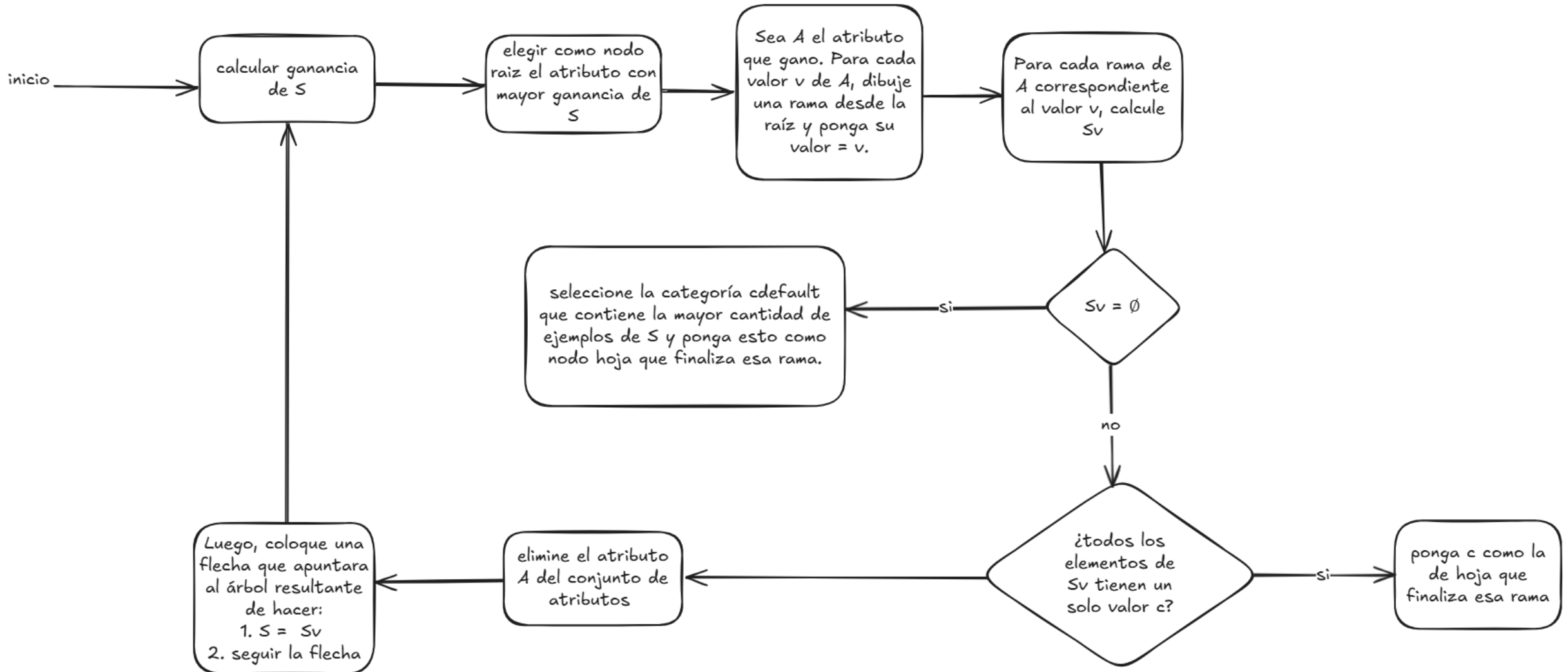
- ID₃ construye un árbol de decisión desde un conjunto de ejemplos dado.
- Se usa este árbol para clasificar nuevos ejemplos.
- El ejemplo tiene varios atributos y pertenece a una clase.
- Los nodos hoja del árbol de decisión contienen el nombre de la clase, mientras que un nodo no hoja es un nodo de decisión.
- El nodo de decisión es una prueba de atributo, donde cada rama representa un posible valor del atributo.
- ID₃ utiliza la ganancia de información para decidir qué atributo se incluye en un nodo de decisión.

Algoritmo

Dado un conjunto de ejemplos S clasificados en categorías c_i entonces:

1. Elija el nodo raíz como el atributo A con la ganancia mas alta con relacion a S .
2. Para cada valor v que A pueda tomar dibuje una rama desde el nodo.
3. Para cada rama de A correspondiente al valor v , calcule S_v (Subconjunto de S donde A tiene el valor v). entonces:
 - 3.1 si S_v es vacio, seleccione la clase c_{default} que contiene la mayor cantidad de ejemplos de S , y ponga esto como la clase de nodo hoja que finaliza esa rama.
 - 3.2 Si S_v contiene solo ejemplos de una sola clase c , entonces ponga c como la clase de nodo hoja que finaliza esa rama.
 - 3.3 De lo contrario, elimine A del conjunto de atributos que se pueden colocar en los nodos. Luego, coloque un nuevo nodo en el árbol de decisión, donde el nuevo atributo que se está probando en el nodo es el que tiene el puntaje más alto en ganancia de información en relación con S_v (nota: no relativa a S). Este nuevo nodo inicia el ciclo nuevamente (desde 2), con S reemplazado por S_v en los cálculos y el árbol se construye iterativamente de esta manera. El algoritmo finaliza cuando se han agotado todos los atributos o el árbol de decisión clasifica perfectamente los ejemplos.

Diagrama de Flujo



Ejemplo: juego de tenis

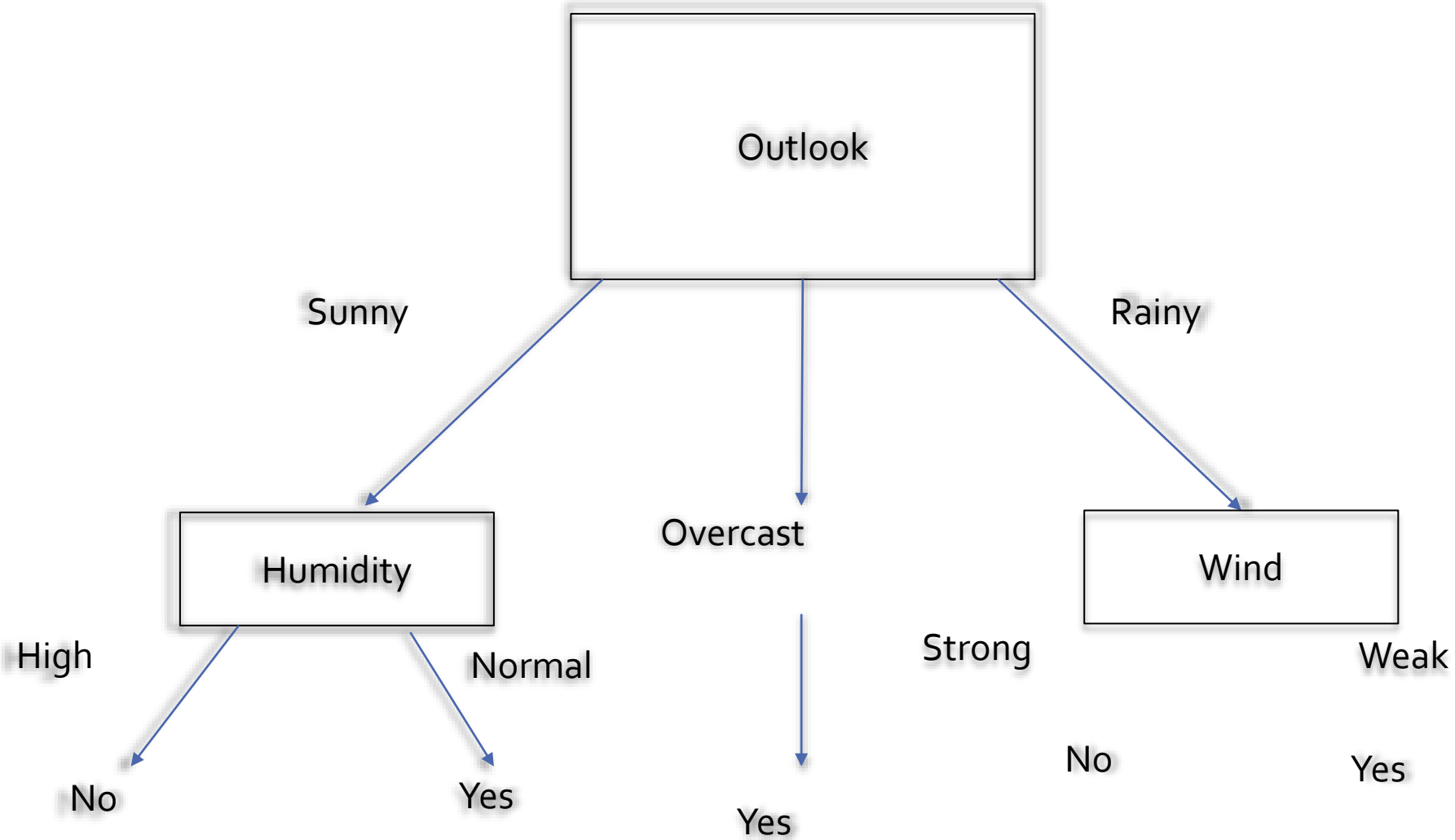
Suponga 14 observaciones sobre el juego de tenis, donde se decide si se juega o no dadas las condiciones meteorológicas de sol, temperatura, humedad, viento; en general tenemos los atributos:

- Outlook
- Temperatura
- Humedad
- Viento
- Decisión (se juega o no: yes, no)

Datos

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Al final del algoritmo obtendremos el siguiente árbol



Entropía y ganancia

Para aplicar el algoritmo necesitamos calcular la entropía y la ganancia.

A continuación se explican estos cálculos.

Entropía

- La entropía es una medida de la aleatoriedad de la información.
- Si el ejemplo es completamente homogéneo la entropía es cero.
- Si la muestra se divide equitativamente, entonces tiene una entropía de uno.
- La entropía puede calcularse así:

$$Entropy(S) = \sum - p(i) \log_2 p(i)$$

Donde $p(i)$ es el porcentaje de S que pertenece a la clase i .
Tenga en cuenta que $S = \{\text{conjunto de muestras completo}\}$.

Ganancia de Información

La ganancia cuantifica **cuánta información aporta un atributo** para clasificar correctamente los datos.

La ganancia de información se puede calcular como :

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum ((|S_v| / |S|) * \text{Entropy}(S_v))$$

Donde:

\sum suma para cada valor de v de todos los posibles valores de atributo A

S_v = subconjunto de S para el que el atributo A vale v

$|S_v|$ = numero de elementos en S_v

$|S|$ = numero de elementos in S

Queremos el árbol de decision para saber si se juega tenis usando datos de 14 días.

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High JAFH	Strong	No

Midiendo la Entropía de S

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

$$Entropy(S) = \sum -p(i) \log_2 p(i)$$

Nuestros datos se clasifican según los posibles valores del campo decisión o sea i esta en $\{yes, no\}$

$$Entropy(S) = -p(Yes) \cdot \log_2 p(Yes) - p(No) \cdot \log_2 p(No)$$

$$Entropy(S) = -(9/14) \cdot \log_2(9/14) - (5/14) \cdot \log_2(5/14)$$

$$= 0.940$$

Obteniendo ganancia

A continuación por cada campo o atributo(Outlook, temp, humidity, wind) calculamos la ganancia.

Empezamos por ejemplo con el campo wind.

Del conjunto S de 14 ejemplos para el atributo viento(wind) tenemos:

- La clasificación de esos 14 ejemplos son 9 YES y 5 NO.
- Los valores de wind pueden ser: *Weak* o *Strong*.

Para calcular la ganancia de wind: Dividimos la tabla original (de 14 datos) en dos subconjuntos de acuerdo a los posibles valores del campo Wind (strong, weak)

Sub conjunto Strong de wind

Day	Outlook	Temp.	Humidity	Wind	Decision
2	Sunny	Hot	High	Strong	No
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
14	Rain	Mild	High	Strong	No

- Hay 6 instances para strong wind. Con 3 yes y 3 no en el campo decision.

$$\text{Entropy}(S, \text{Wind}=\text{Strong}) = - (3/6) \cdot \log_2(3/6) - (3/6) \cdot \log_2(3/6) \\ = 1$$

Sub conjunto Weak

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
13	Overcast	Hot	Normal	Weak	Yes

Hay 8 instancias para weak wind. Con 2 no en el campo Decisión y 6 ítems son **yes**.

Calculamos la entropía para el conjunto Weak

$$\begin{aligned}\text{Entropy}(S, \text{Wind}=\text{Weak}) &= -(2/8) \cdot \log_2(2/8) - (6/8) \cdot \log_2(6/8) \\ &= \mathbf{0.811}\end{aligned}$$

Calculando la ganancia

Calculamos ahora la ganancia de cada grupo

- Weak
- Strong

Entropia de cada S_v

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum ((|S_v| / |S|) * \text{Entropy}(S_v))$$

$|S|$ =cardinalidad de $S=14$

$|S_{\text{weak}}|=8$ dado que hay 8 elementos de weak en S

$|S_{\text{strong}}|=6$ dado que hay 6 elementos de strong en S

$$\text{Entropy}(S_v) = \sum -p(i) \log_2 p(i)$$

Donde $p(i)$ es el porcentaje de S_v que pertenece a la clase i ya sea (*Weak* o *Strong*)

$$\text{Entropy}(S_{\text{weak}}) = - (6/8) * \log_2(6/8) - (2/8) * \log_2(2/8) = 0.811$$

$$\text{Entropy}(S_{\text{strong}}) = - (3/6) * \log_2(3/6) - (3/6) * \log_2(3/6) = 1.00$$

Ganancia de cada S_v

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum ((|S_v| / |S|) * \text{Entropy}(S_v))$$

$$\begin{aligned}\text{Gain}(S, \text{Wind}) &= \text{Entropy}(S) - (8/14) * \text{Entropy}(S_{\text{weak}}) - (6/14) * \text{Entropy}(S_{\text{strong}}) \\ &= 0.940 - (8/14) * 0.811 - (6/14) * 1.00 \\ &= 0.048\end{aligned}$$


Para cada atributo de la tabla se calcula la ganancia y se toma la mas alta
Tenemos el de Wind, nos faltan el de OutLook, Temperatura y humedad.

Entropia de cada S_v

Para cada atributo de la tabla se calcula la ganancia y se toma la mas alta.

Tenemos el de Wind, nos faltan el de OutLook, Temperatura y humedad.

Calculando todas las ganancias tenemos

- $\text{Gain}(S, \text{wind}) = 0.048$
 - $\text{Gain}(S, \text{Outlook}) = 0.246$
 - $\text{Gain}(S, \text{Temperature}) = 0.029$
 - $\text{Gain}(S, \text{Humidity}) = 0.151$
- Mayor ganancia
- 

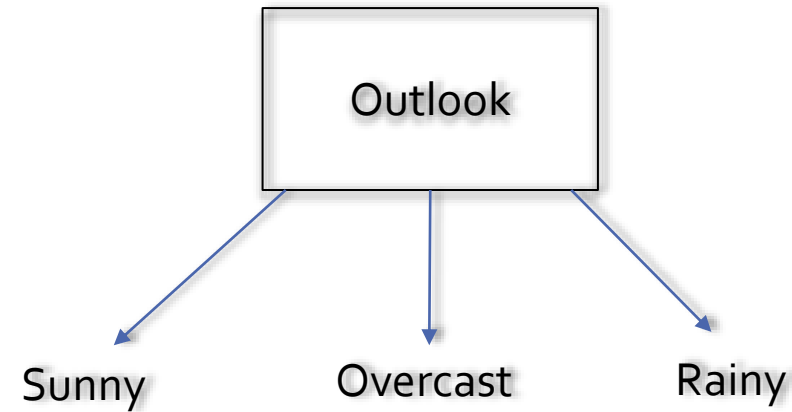
Según el algoritmo pasos 1 y 2:

1. Elija el nodo raíz como el atributo A con la ganancia mas alta con relación a S.
2. Para cada valor v que A pueda tomar dibuje una rama desde el nodo.

Aplicando el algoritmo

Paso 1: Como outlook tiene mayor ganancia lo ponemos como raiz.

Paso 2: como Outlook tiene los valores (sunny, overcast, rainy), creamos esas ramas.



Recursión

Ahora tenemos 3 ramas:

- Sunny
- Overcast
- Rainy

Analizamos cada rama como sigue:

Para la rama Overcast

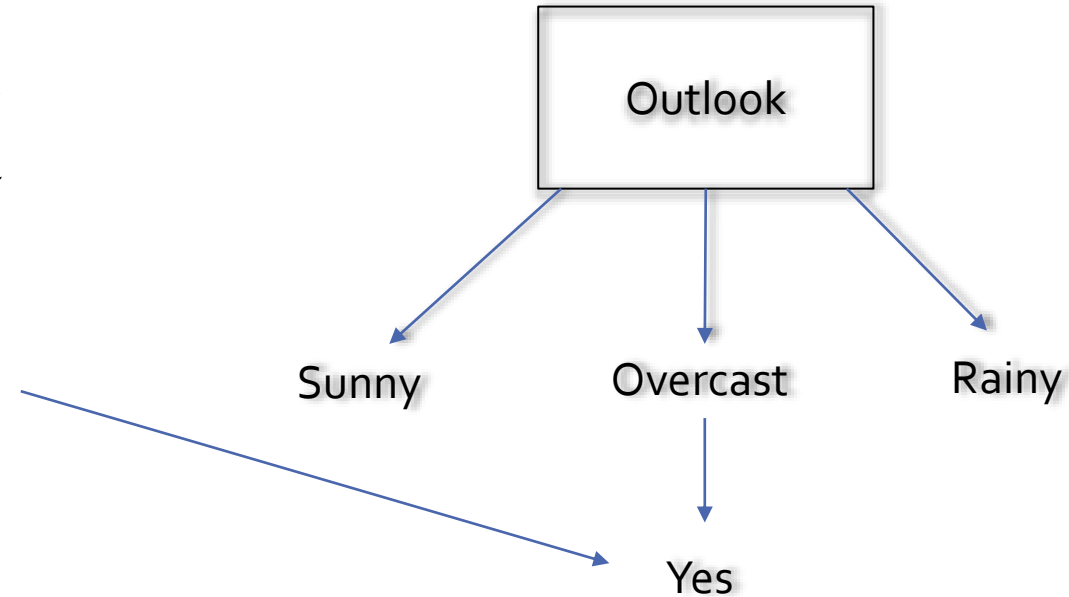
Vea que para el valor overcast en la rama Outlook, el campo decisión siempre será yes.

Day	Outlook	Temp.	Humidity	Wind	Decision
3	Overcast	Hot	High	Weak	Yes
7	Overcast	Cool	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes

Siguiendo el algoritmo

3.2 Si S_v contiene solo ejemplos de una sola categoría c , entonces ponga c como la categoría de nodo hoja que finaliza esa rama.

Aplicando 3.2 del algoritmo a la rama overcast
Ponemos una hoja yes y ahí termina esa rama



Quedan 2 ramas por analizar

- Sunny
- Rainy

Analicemos Sunny

Rama Sunny

Tenemos 5 instancias para sunny el campo Decision tiene la probabilidad

3/5 porcentaje no,

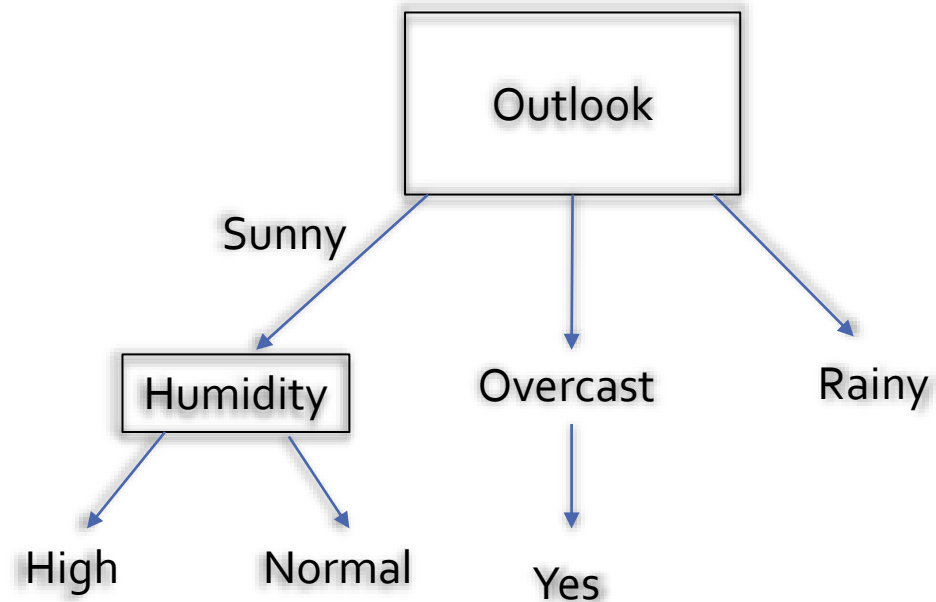
2/5 porcentaje yes

Calculamos la ganancia de los campos

- $\text{Gain}(\text{Outlook}=\text{Sunny}|\text{Temperature}) = 0.570$
- $\text{Gain}(\text{Outlook}=\text{Sunny}|\text{Humidity}) = 0.970$
- $\text{Gain}(\text{Outlook}=\text{Sunny}|\text{Wind}) = 0.019$

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

Como Humidity tiene la mayor ganancia será el inicio de la rama, se agregan 2 ramas (High, Normal) por que son los posibles valores que puede tener Humidity.

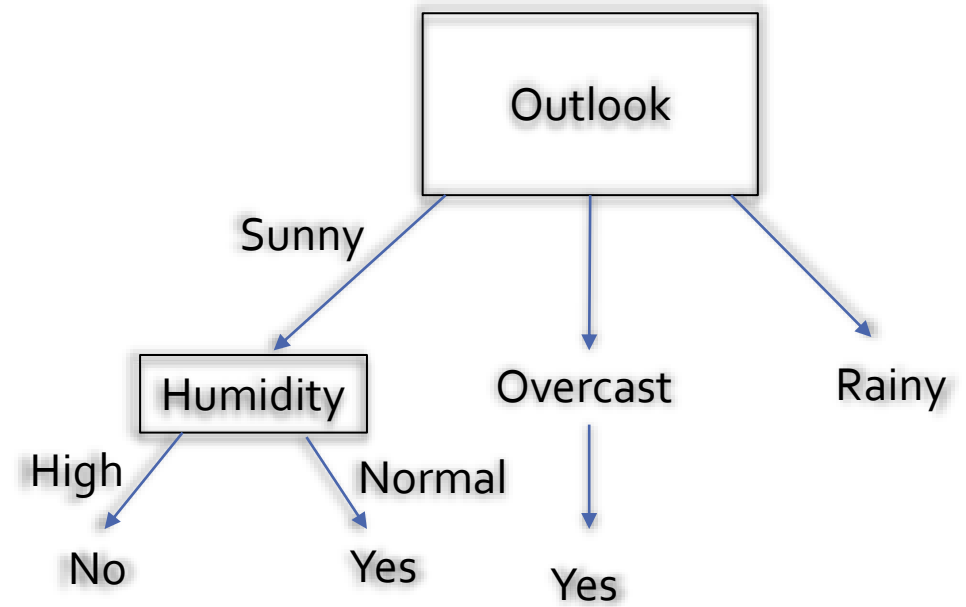


A continuación analicemos las ramas High y Normal

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No

Day	Outlook	Temp.	Humidity	Wind	Decision
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

- El campo Decision será siempre **no** cuando humidity = **high**.
- El campo Decision sera siempre **yes** cuando humidity = **normal**.



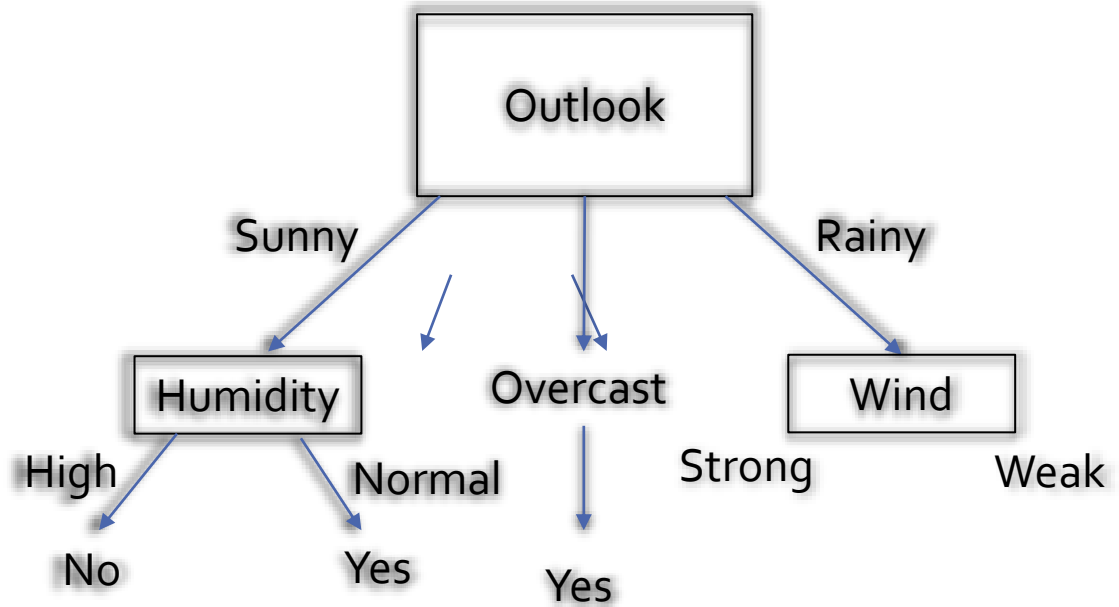
Como High y normal tienen un solo valor tenemos

Analizando el campo Rain

Ganancia de información (gain) para Rain:

- $\text{Gain}(\text{Outlook}=\text{Rain} \mid \text{Temperature}) = 0.02$
- $\text{Gain}(\text{Outlook}=\text{Rain} \mid \text{Humidity}) = 0.02$
- $\text{Gain}(\text{Outlook}=\text{Rain} \mid \text{Wind}) = 0.971$

Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

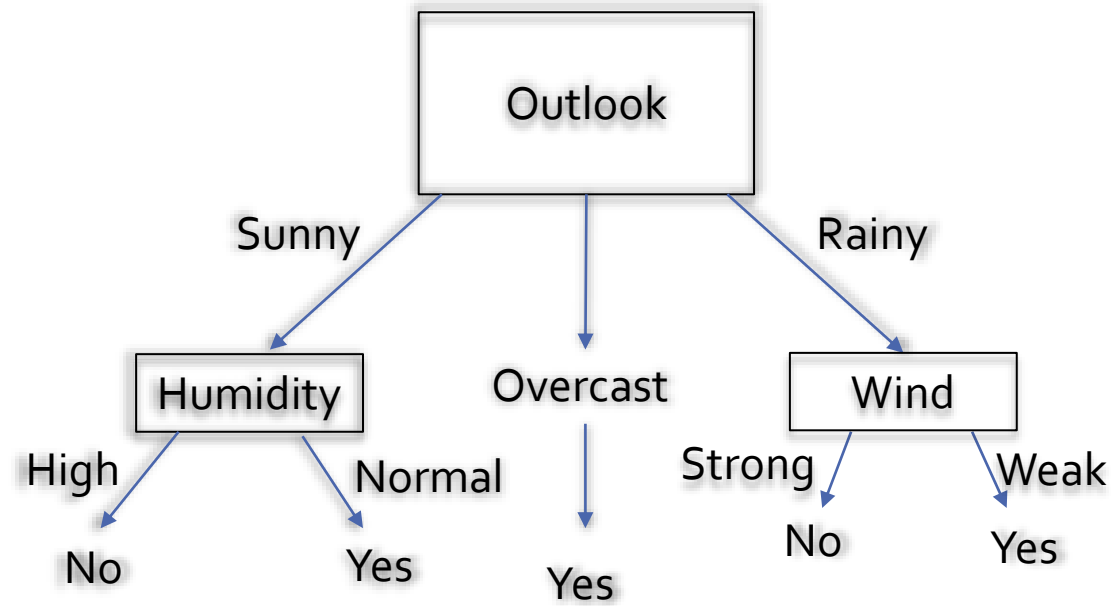


Como wind tuvo la mayor ganancia trabajamos sobre este campo

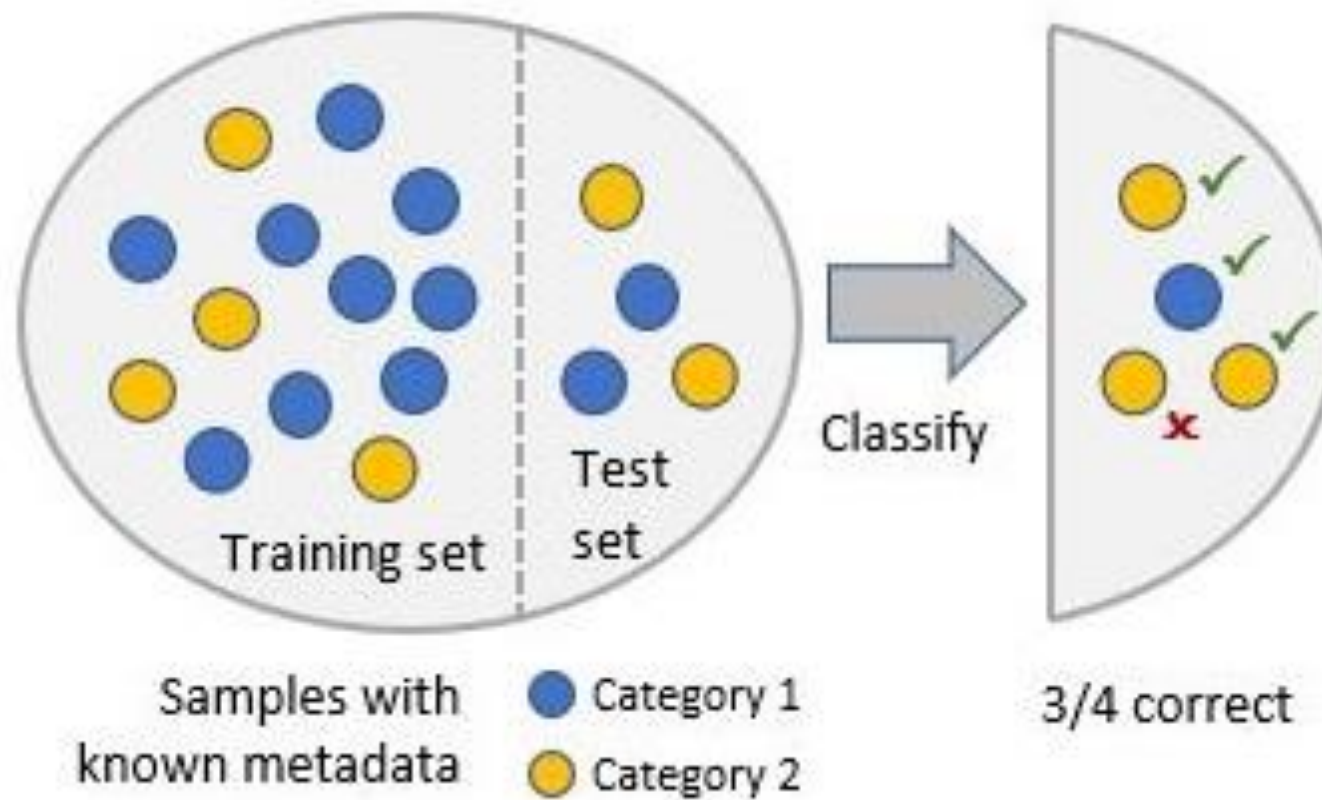
Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes

Day	Outlook	Temp.	Humidity	Wind	Decision
6	Rain	Cool	Normal	Strong	No
14	Rain	Mild	High	Strong	No

- El campo Decisión será siempre **yes** si wind = **weak** y outlook = **rain**.
- El campo Decisión será siempre **no** si wind = **strong** y outlook = **rain**.



Crossvalidation



Validación cruzada

El **10-Fold Cross-Validation** (Validación Cruzada de 10 Pliegues o 10 Iteraciones) es una técnica **fundamental** en Machine Learning para evaluar el rendimiento y la generalización de un modelo predictivo.

La Idea Central en Términos Sencillos

Imagina que tienes un conjunto de datos con 1000 ejemplos (instancias). Necesitas entrenar un modelo y saber qué tan bien funcionará con datos nuevos que nunca ha visto.

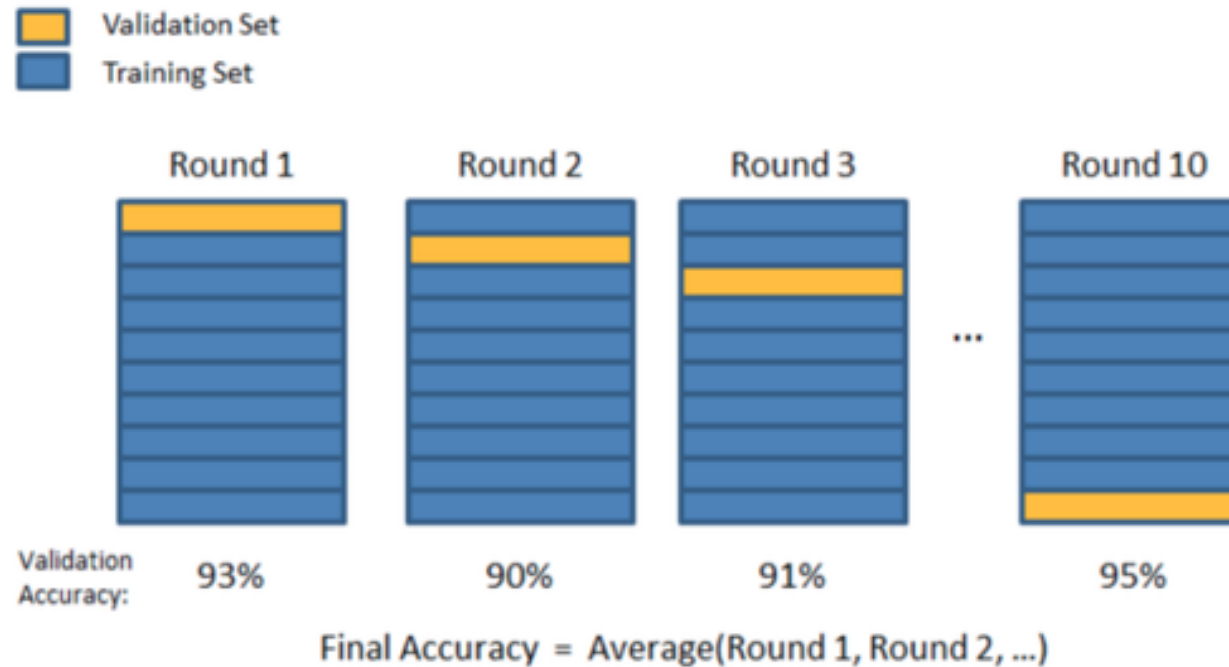
- **El error grave:** Entrenar el modelo con *todos* los 1000 datos y luego probarlo con esos *mismos* 1000 datos. El modelo tendrá un rendimiento artificialmente alto porque solo se evaluó a sí mismo, no su capacidad para generalizar. Esto se llama **evaluación optimista**.
- **La solución simple:** Dividir los datos en dos grupos: 700 para entrenar y 300 para probar. Esto es mucho mejor, pero su desventaja es que el rendimiento final depende mucho de *qué* 300 ejemplos se eligieron para la prueba. Quizá esos 300 eran muy fáciles o muy difíciles.

El 10-Fold Cross-Validation soluciona estos problemas de una manera elegante y robusta.

Validación cruzada

- ¿Cómo Funciona Paso a Paso?
- 1. **Dividir:** Tu conjunto de datos completo se divide aleatoriamente en **10 subconjuntos (o "pliegues")** de aproximadamente el mismo tamaño. Por ejemplo, 1000 datos se dividen en 10 grupos de 100.
- 2. **Iterar 10 Veces:** El proceso de entrenamiento y prueba se repite **10 veces**. En cada una de las 10 iteraciones:
 1. **Un pliegue diferente** se usa como **conjunto de prueba** (los 100 datos de color amarillo).
 2. **Los otros 9 pliegues** se combinan para formar el **conjunto de entrenamiento** (los 900 datos de color azul).
 3. Se entrena el modelo desde cero con el conjunto de entrenamiento.
 4. Se prueba el modelo entrenado en el conjunto de prueba y se guarda una métrica de rendimiento (por ejemplo, el porcentaje de aciertos o "accuracy").
- 3. **Resultado Final:** Al final de las 10 iteraciones, tienes **10 medidas de rendimiento** (una por cada iteración). El rendimiento reportado del modelo es el **promedio** de estas 10 medidas.

10 fold crossvalidation



Actividad

- Crear un árbol de decisión usando el algoritmo ID3 para los datos de la flor iris: <https://archive.ics.uci.edu/dataset/53/iris>.
- Contrastar con un árbol de decisión creado usando la aplicación Weka (apóyese en las siguientes diapositivas)

Weka

- **Weka** es una suite de software libre (bajo licencia GNU GPL) para el **aprendizaje automático** (*machine learning*) y la **minería de datos** (*data mining*). Su nombre es un acrónimo de **Waikato Environment for Knowledge Analysis** (Entorno Waikato para el Análisis de Conocimiento), ya que fue desarrollado en la Universidad de Waikato en Nueva Zelanda.
- Para hacer una analogía sencilla, **Weka es como un "Word" o "Excel" para la ciencia de datos**, pero especializado en algoritmos de inteligencia artificial y análisis predictivo.

Descargar:

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

Weka Características principales

1. Interfaz Gráfica Amigable (GUI)

2. Colección Extensa de Algoritmos: Incluye algoritmos listos para usar en tareas como:

1. **Preprocesamiento de datos:** Filtrado, normalización, manejo de valores faltantes, etc.
2. **Clasificación:** Árboles de decisión (J48, que es una implementación de C4.5), regresión logística, Support Vector Machines (SVM), Naive Bayes, entre muchos otros.
3. **Regresión:** Predecir valores numéricos.
4. **Agrupamiento (Clustering):** Algoritmos como K-Means para encontrar grupos naturales en los datos.
5. **Reglas de Asociación:** Para descubrir relaciones entre variables (como el famoso algoritmo Apriori).
6. **Selección de atributos:** Para identificar las características más relevantes de tu dataset.

3. Herramientas de Visualización: Permite visualizar los datos (gráficos de dispersión, histogramas) y los resultados de los modelos (como el árbol de decisiones generado o los clusters encontrados) para una mejor interpretación.

4. Entorno de Experimentación: Permite comparar de manera sistemática el rendimiento de diferentes algoritmos en los mismos datos, generando métricas de evaluación como precisión, matriz de confusión, etc.

5. Modo Consola/Línea de Comandos: Para usuarios avanzados, ofrece la posibilidad de ejecutar todas sus funciones desde la terminal, lo que es útil para automatizar tareas.

6. API en Java: Está escrito completamente en Java, lo que significa que es multiplataforma (funciona en Windows, macOS, Linux) y permite a los desarrolladores integrar sus algoritmos en sus propias aplicaciones Java.

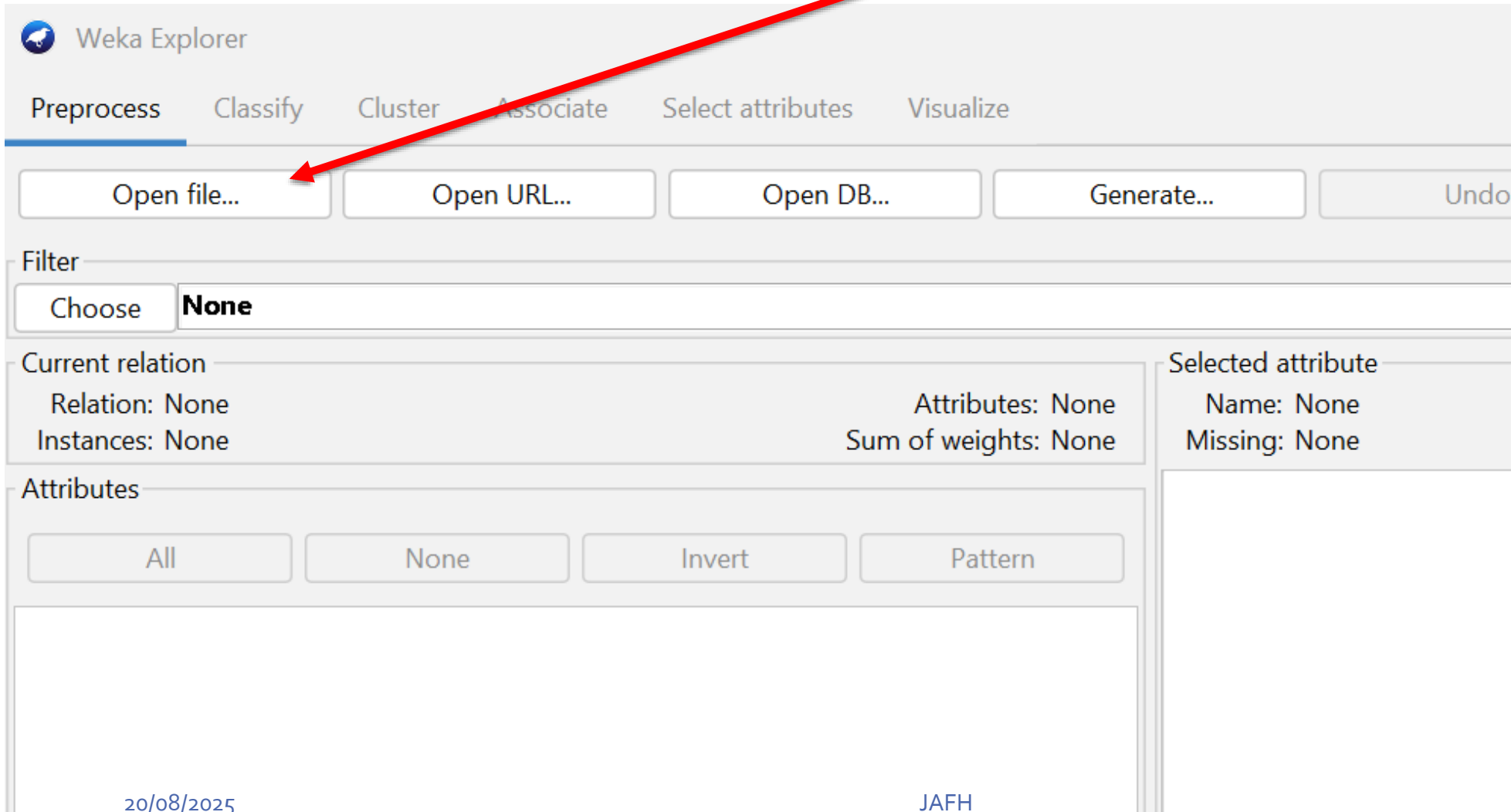
Weka: usar Weka



Iniciar la aplicación y dar clic en Explorer.

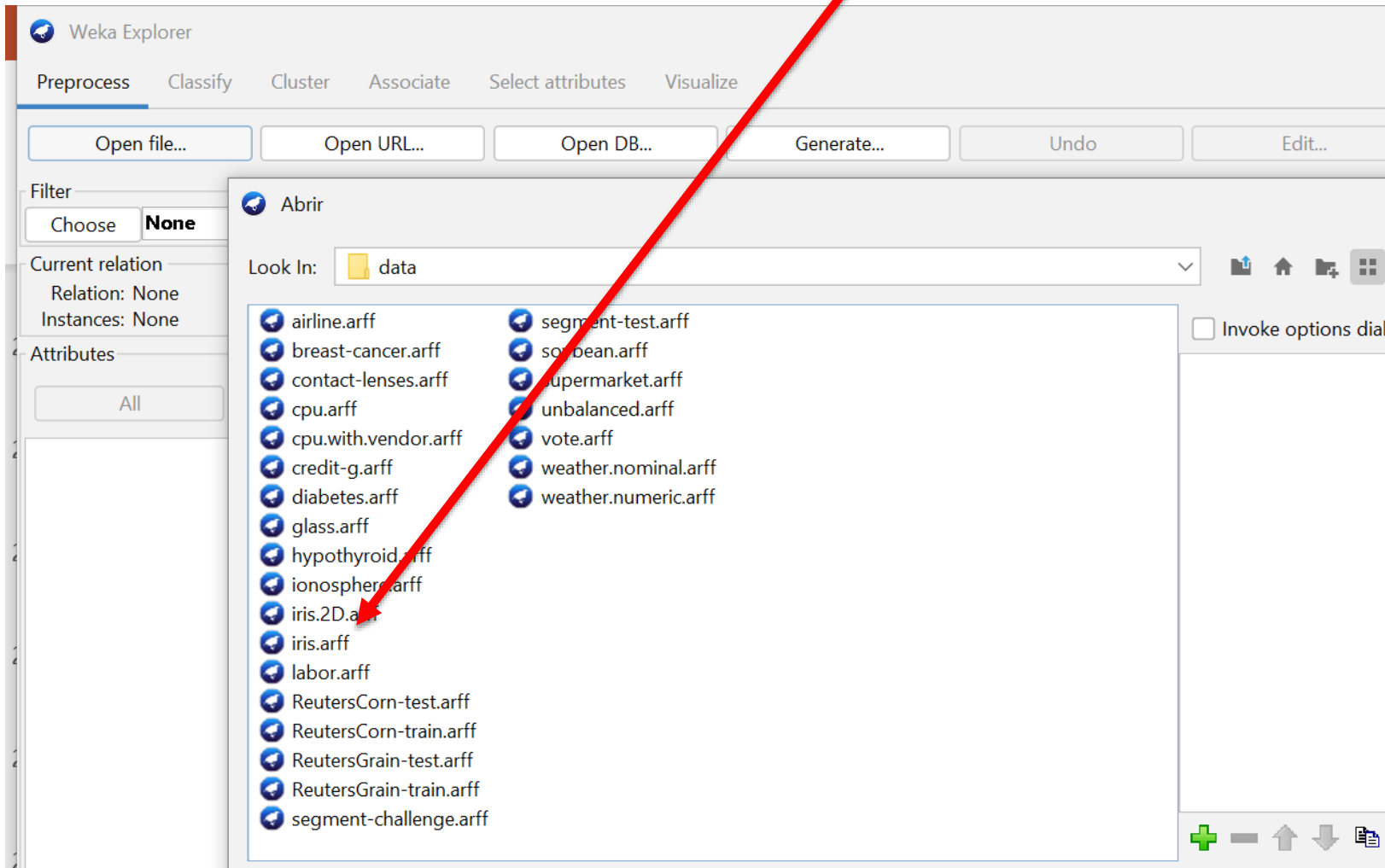
Weka

Abrir archivo de datos, debe ser *.arff



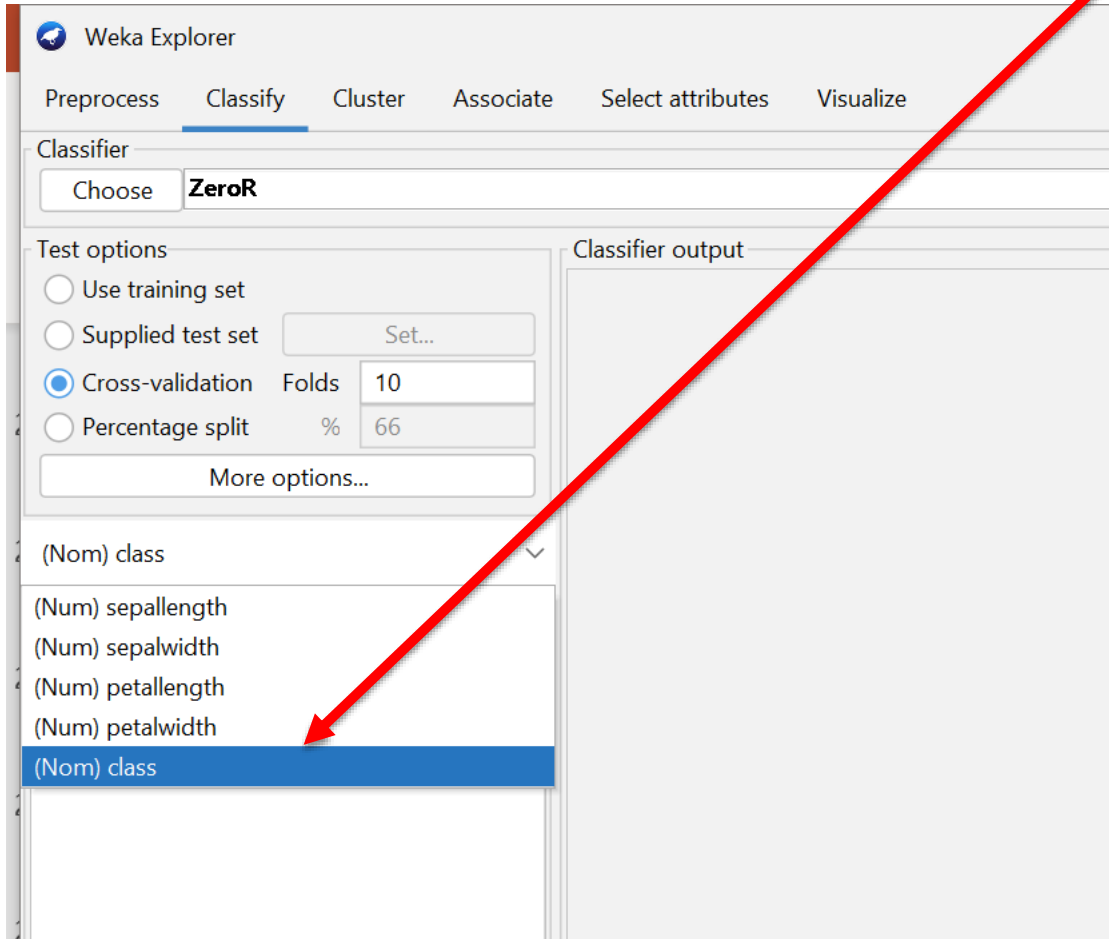
Weka

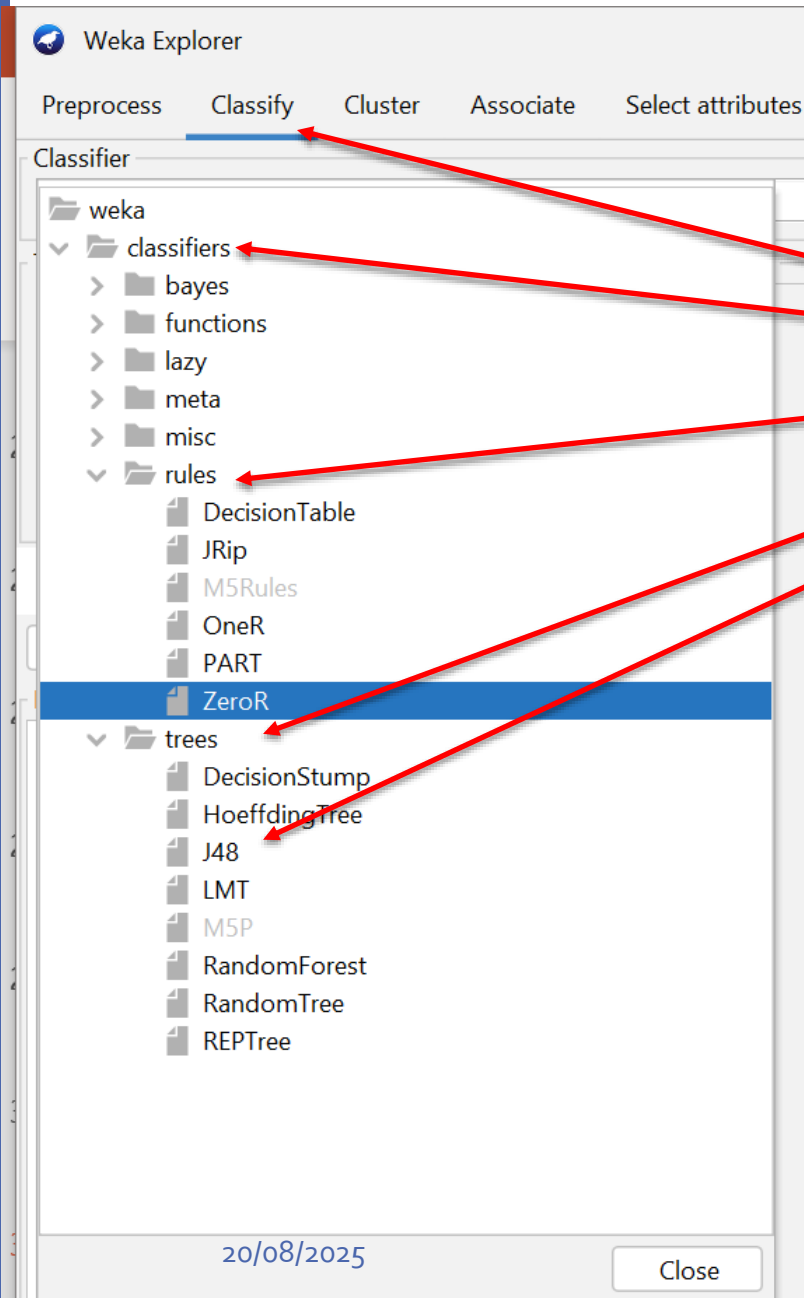
Seleccionar el conjunto de datos iris
Ruta: C:\Program Files\Weka-3-8-6\data



Weka

Seleccionar el atributo de la clase,
atributo por cual se hará la
clasificación

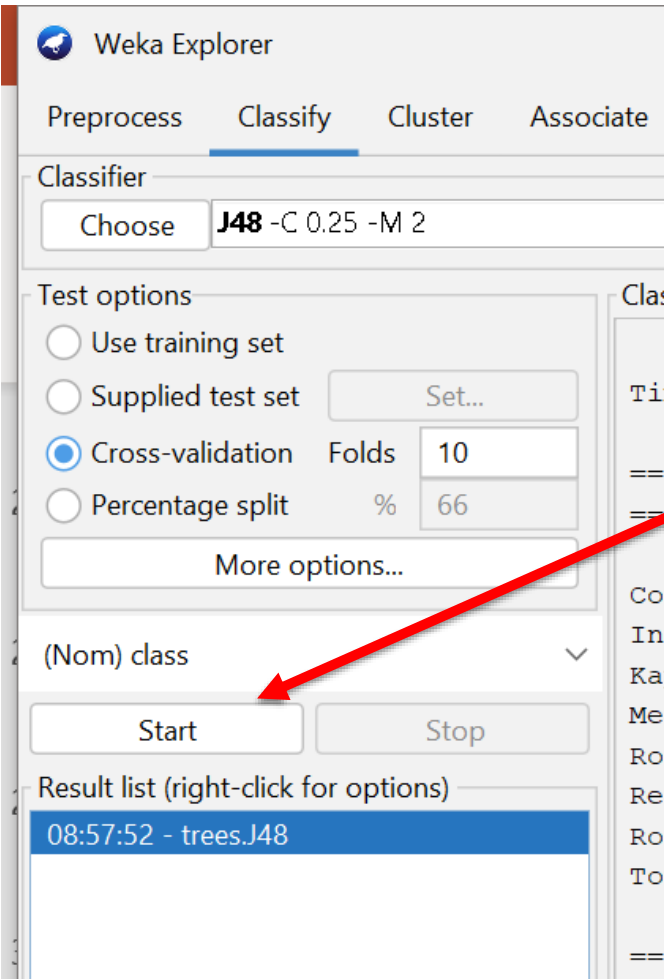




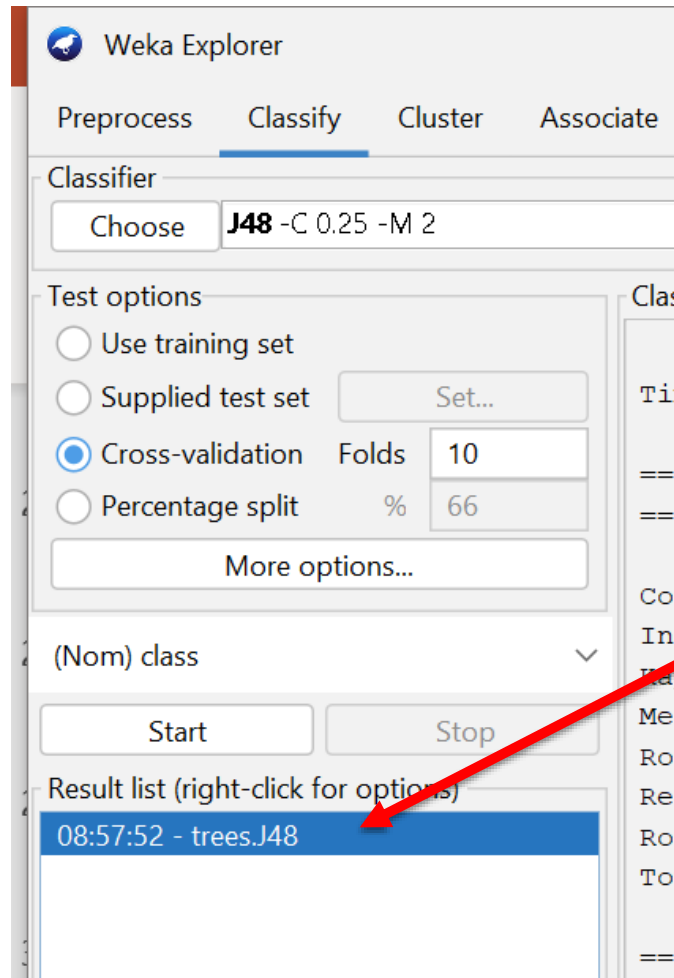
Seleccionar algoritmo de clasificación:

Clasificación/rules/tree/j48

Que es una versión mejorada de ID3

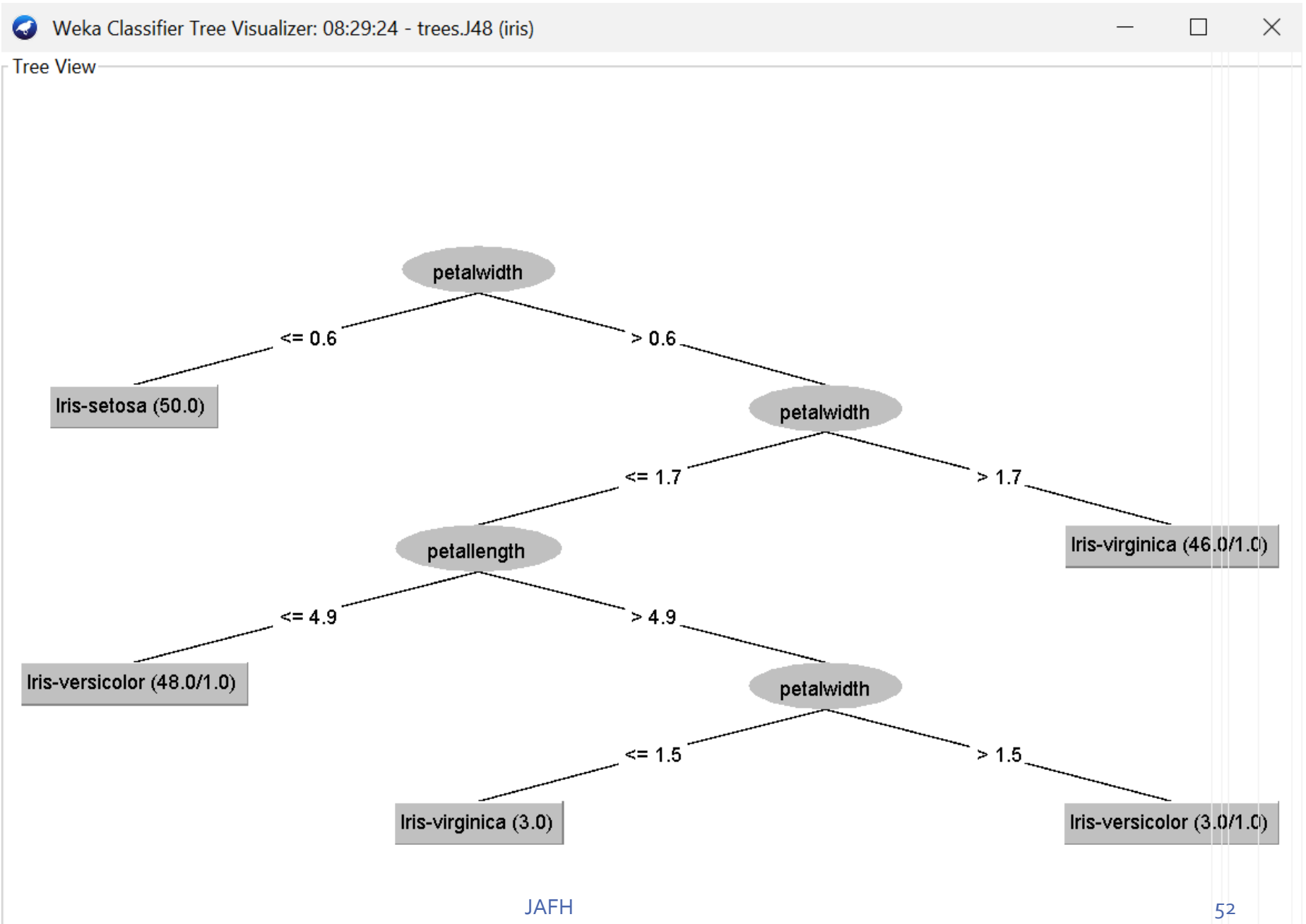


Seleccionar: Start



Clic derecho y seleccionar:
visualizar árbol

Árbol



Fin

JAFH

Ligas:

1. https://waikato.github.io/weka-wiki/downloading_weka/
2. <https://archive.ics.uci.edu/dataset/53/iris>.

Herramientas de apoyo recomendadas:

- Excell
- Excalidraw