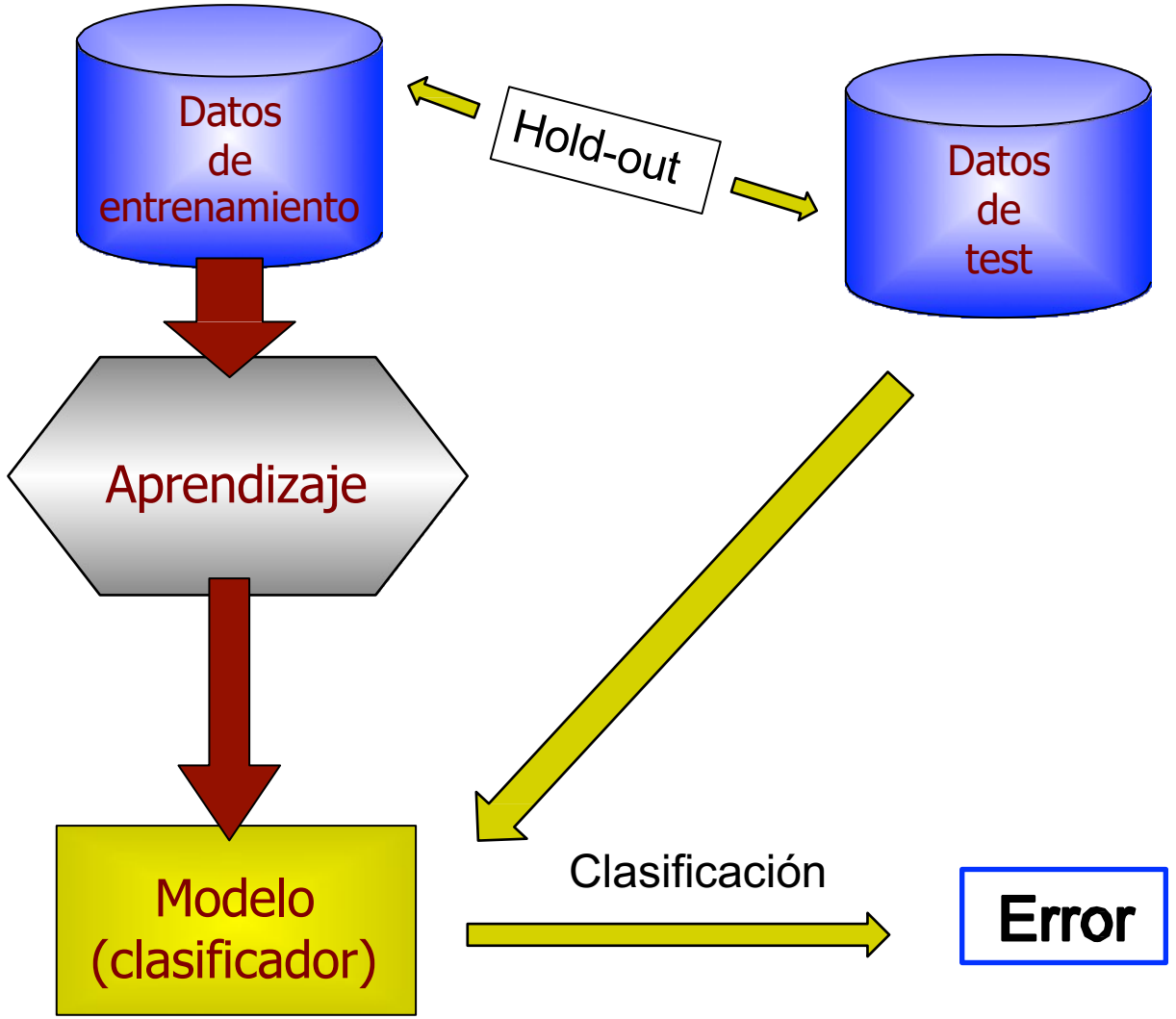
TEMA 8. TÉCNICAS DE CLASIFICACIÓN

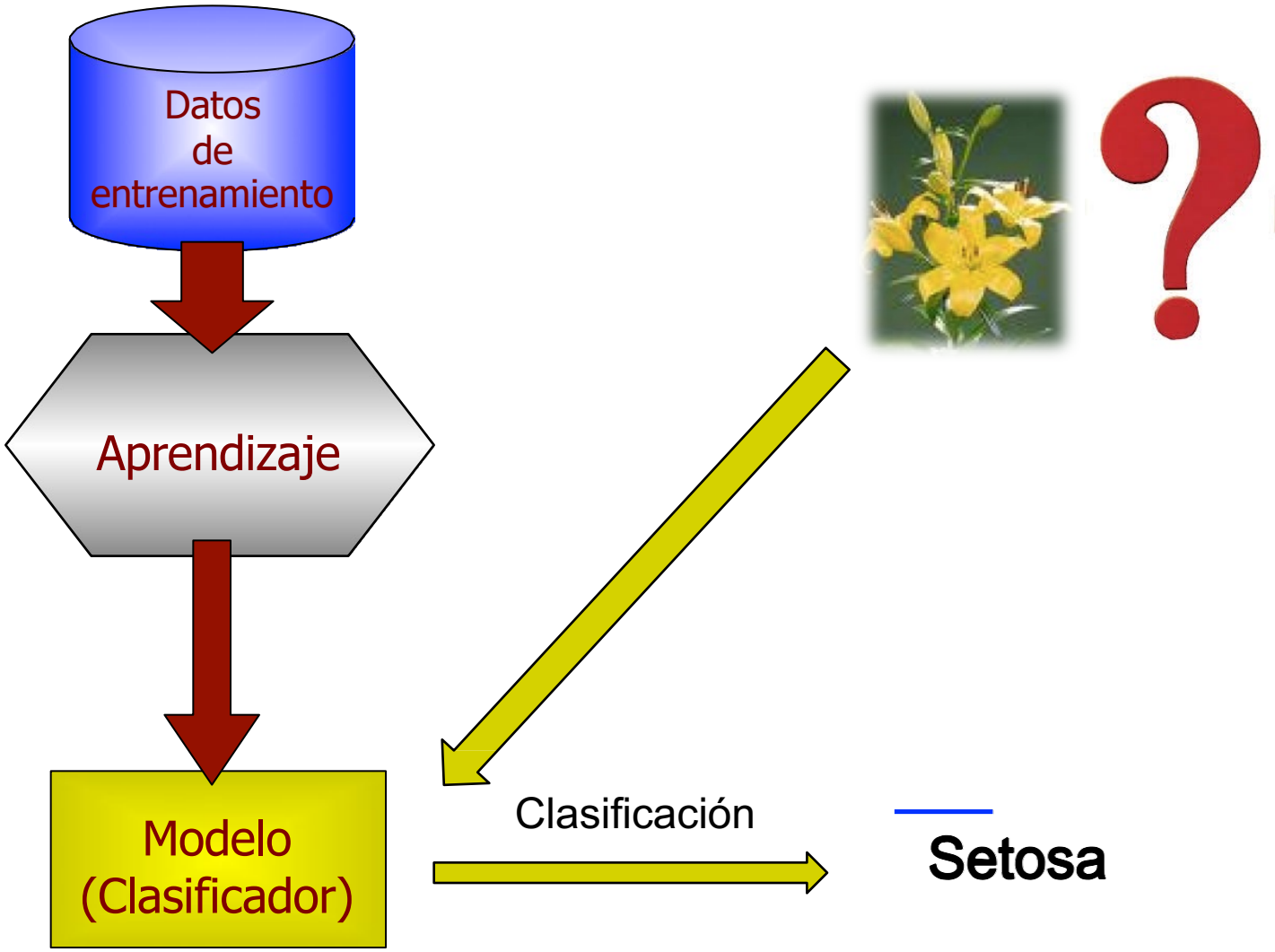
Contenidos

1. Introducción
2. Clasificadores bayesianos
3. Árboles de decisión
4. Vecinos más cercanos
5. Máquinas de vectores soporte
6. Ejercicios de clasificación

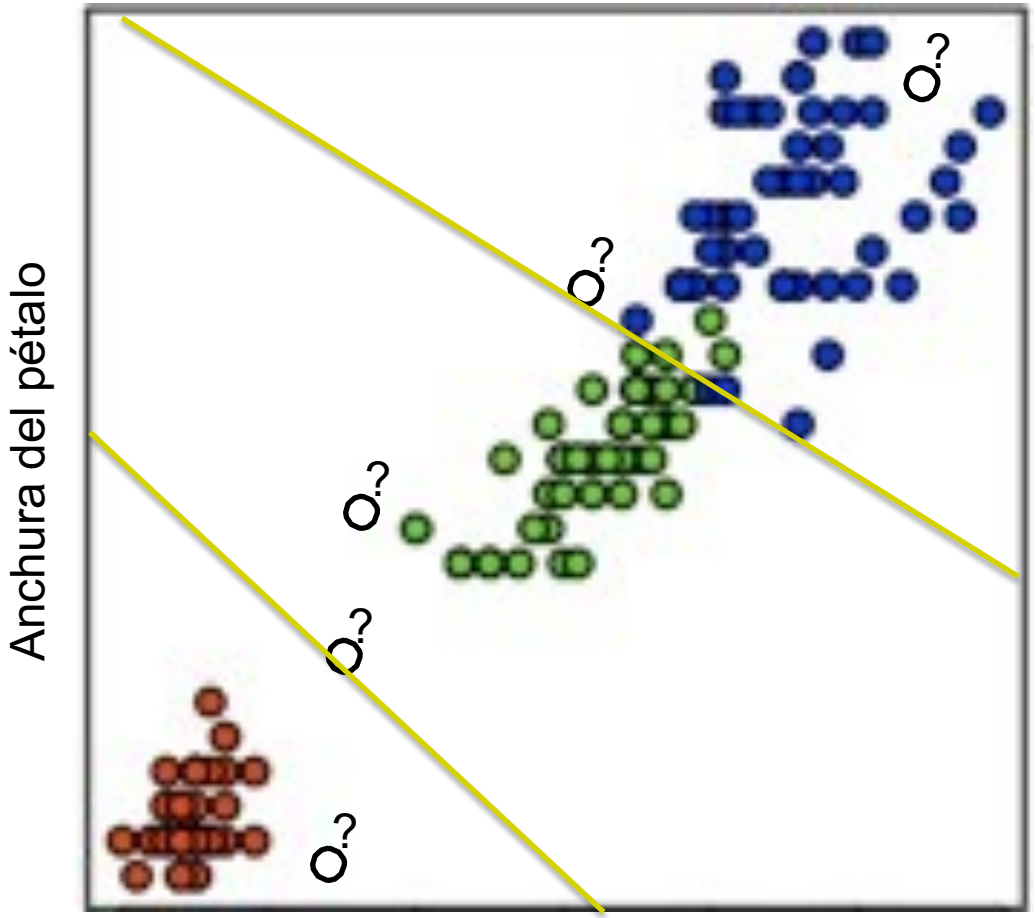
1. Introducción



1. Introducción



1. Introducción



Setosa

versicolor

virginica

1. Introducción

Validación de la clasificación

- **Hold-out**: Dividir datos en entrenamiento y test
  - Solución simple que puede ser usada si tenemos muchos datos (etiquetados)
  - Sin embargo, los datos (etiquetados) normalmente son limitados. Se necesitan técnicas más sofisticadas.
- **K-fold cross-validation**: Se generan diferentes conjuntos de entrenamiento y de test
- **Leave-one-out**: Todas las instancias se usan en el entrenamiento excepto una que se usa en el test

1. Introducción

Validación de la clasificación

Hold-out

- Reserva una determinada cantidad para testear y el resto lo usa para entrenar (una tercera parte para test, el resto para entrenar)
- **Conjunto de Test:** instancias independientes que no se han usado para obtener el modelo
  Mientras más grande sea el conjunto de entrenamiento mejor es el modelo
  Mientras más grande sea el conjunto de test más aproximada es la estimación del error
- **Problemas**:
  Que las muestras no sean representativas
  Número de instancias bajo, por tanto, conjunto test muy pequeño

1. Introducción

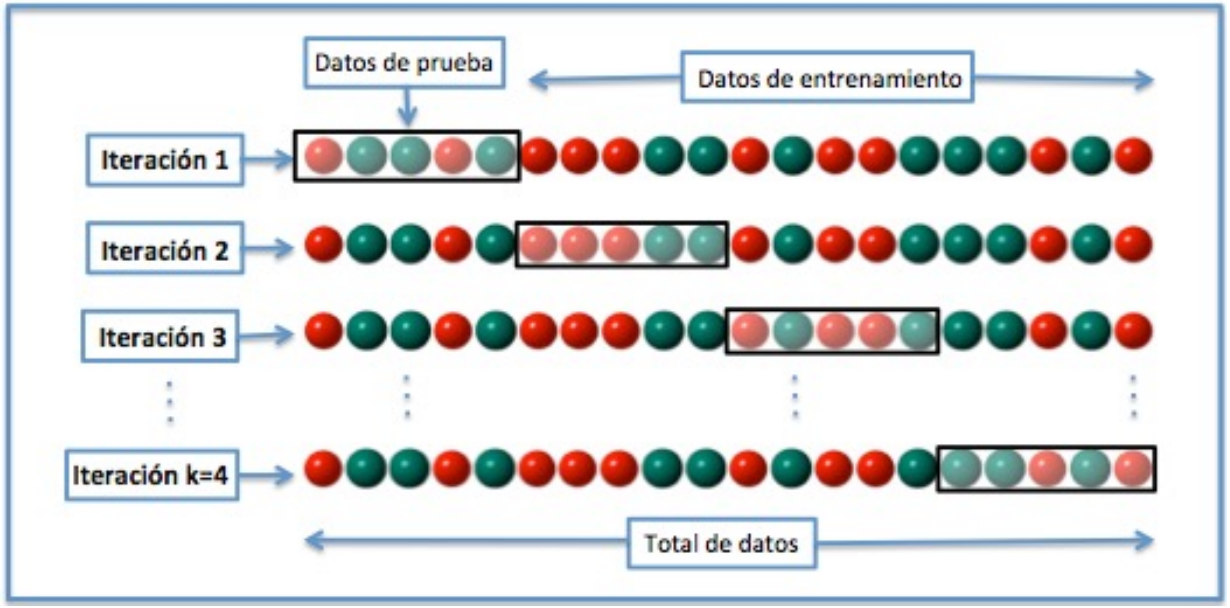
Validación de la clasificación

K-fold Cross-validation

- Evita que los conjuntos de tests se solapen (normalmente k = 10)
  - Primer paso: dividir datos en *k* subconjuntos de igual tamaño
  - Segundo paso: usar cada subconjunto como test y el resto como entrenamiento
- El error se calcula haciendo la media de los *k* errores
- **Caso particular: Leave-One-Out**
  - El número de bolsas es el número de instancias en el conjunto de entrenamiento
  - Si el número de instancias es N, se construyen N clasificadores o regresores
  - Muy costoso computacionalmente

1. Introducción

<u>Validación de la clasificación</u>: K-fold Cross-validation



1. Introducción

Evaluación de la clasificación binaria

		Predicción	
		P	**N**
Clase real	**P**	**TP**: True positive	**FN**: False negative
	N	**FP**: False positive	**TN**: True negative

Exactitud del clasificador:
Accuracy =
(TP+TN) / (TP+TN+FP+FN)

Limitaciones:

- Supongamos un problema con 2 clases: 9990 ejemplos de la clase 1 y 10 ejemplos de la clase 2
- Si el modelo de clasificación siempre dice que los ejemplos son de la clase 1, su precisión es 9990/10000 = 99.9%

1. Introducción

Evaluación de la clasificación binaria

Sensibilidad (recall, cobertura positiva): Sens = TP / (TP+FN)

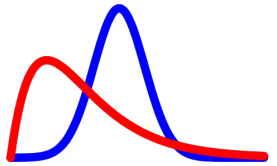
Especificidad (cobertura negativa): Espec = TN / (TN+FP)

Valor predictivo positivo (precisión): Prec = PPV = TP / (TP+FP)

Valor predictivo negativo: NPV = TN / (TN+FN)

F-measure (F1): F1 = 2 * PPV * Sens / (PPV + Sens)

2. Clasificadores bayesianos



- Se basan en probabilidades.
- Por ejemplo, en el dataset “weather”, si el test es:

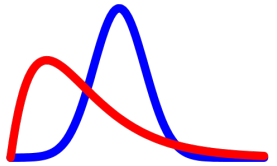
  <sunny, cool, high, strong>

  - ¿Qué probabilidad es mayor la de jugar al tenis o la de no jugar al tenis?
- **P(jugar al tenis)** =

  P(sunny/Yes)*P(cool/Yes)*P(high/Yes)*P(strong/Yes)*P(Yes)
- **P(no jugar al tenis)** =

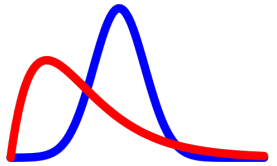
  P(sunny/No)*P(cool/No)*P(high/No)*P(strong/No)*P(No)

2. Clasificadores bayesianos



Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

2. Clasificadores bayesianos



P(sunny/YES)=2/9
P(cool/YES)=3/9
P(high/YES)=3/9
P(strong/YES)=3/9
P(YES)=9/14

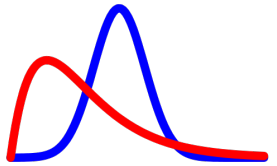
P(sunny/NO)=3/5
P(cool/NO)=1/5
P(high/NO)=4/5
P(strong/NO)=3/5
P(NO)=5/14

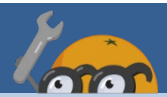
- Probabilidad de jugar =0,0053
- Probabilidad de no jugar=0,0206

Luego lo más probable es que la etiqueta sea NO

Método Naïve Bayes es el más conocido

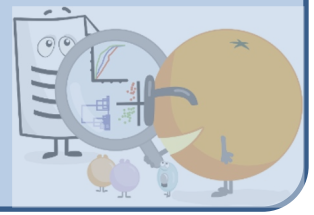
2. Clasificadores bayesianos



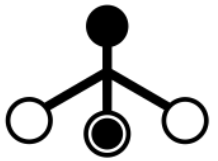


Práctica con Orange

- **Datasets de estudio**
  - Iris
- **Algoritmo**
  - Naive Bayes
- **Métricas de evaluación**
  - CA
  - Precision
  - Recall
  - F-measure (F1)
- **Validación**
  - Validación cruzada con 10 bolsas



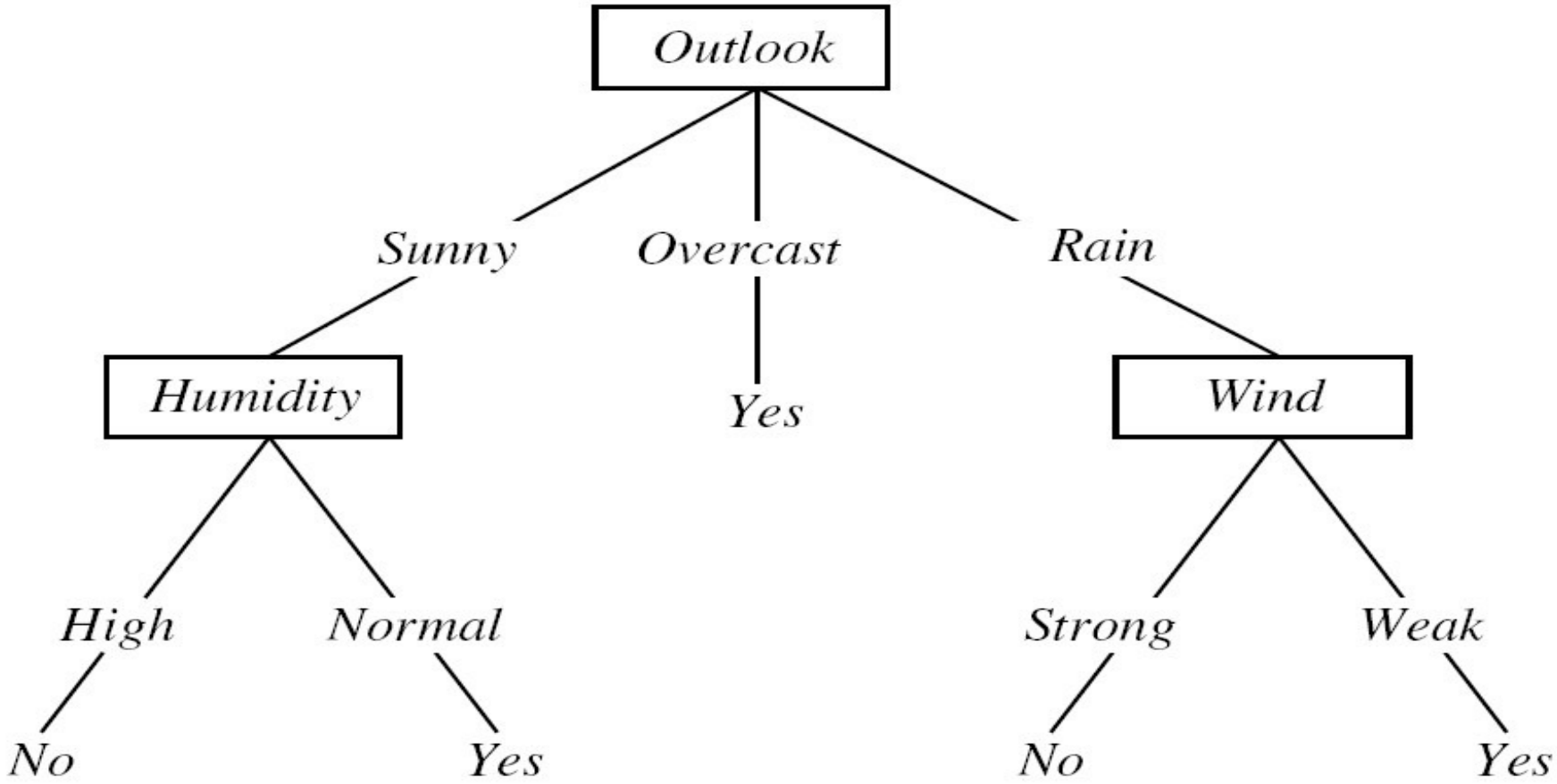
3. Árboles de decisión



Example	Outlook	Temperature	Humidity	Wind	PlayTennis
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

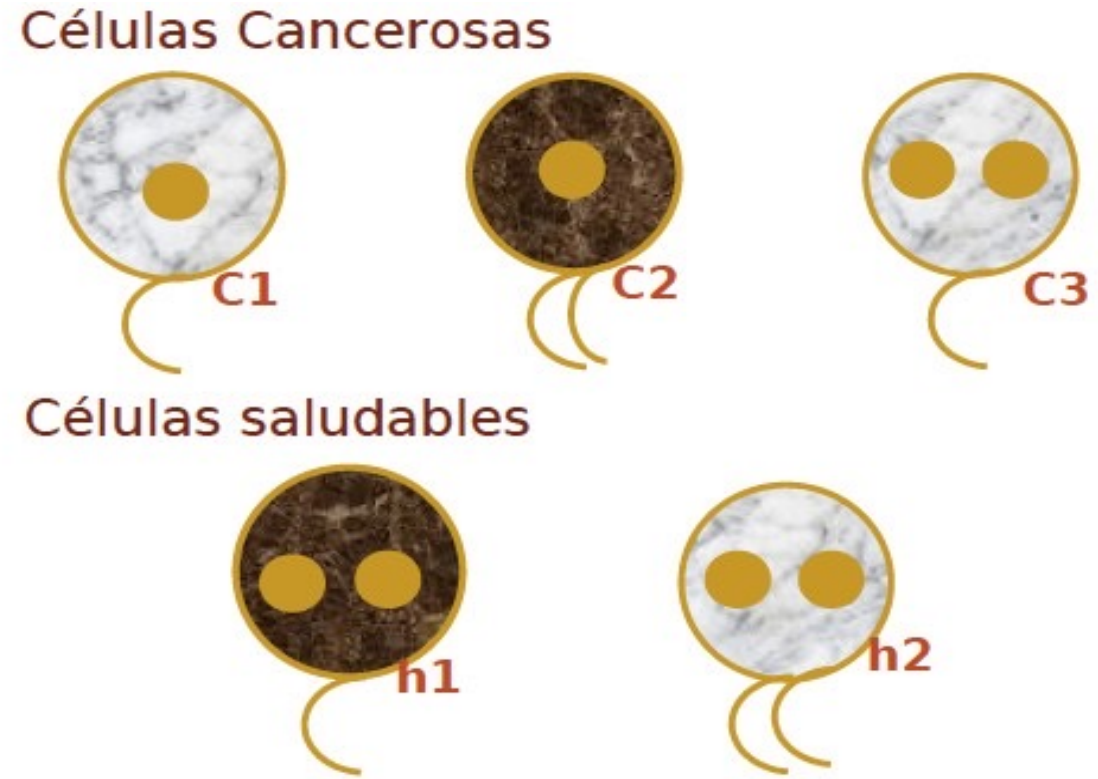
3. Árboles de decisión

Ejemplo 1: ¿Jugar al tenis?

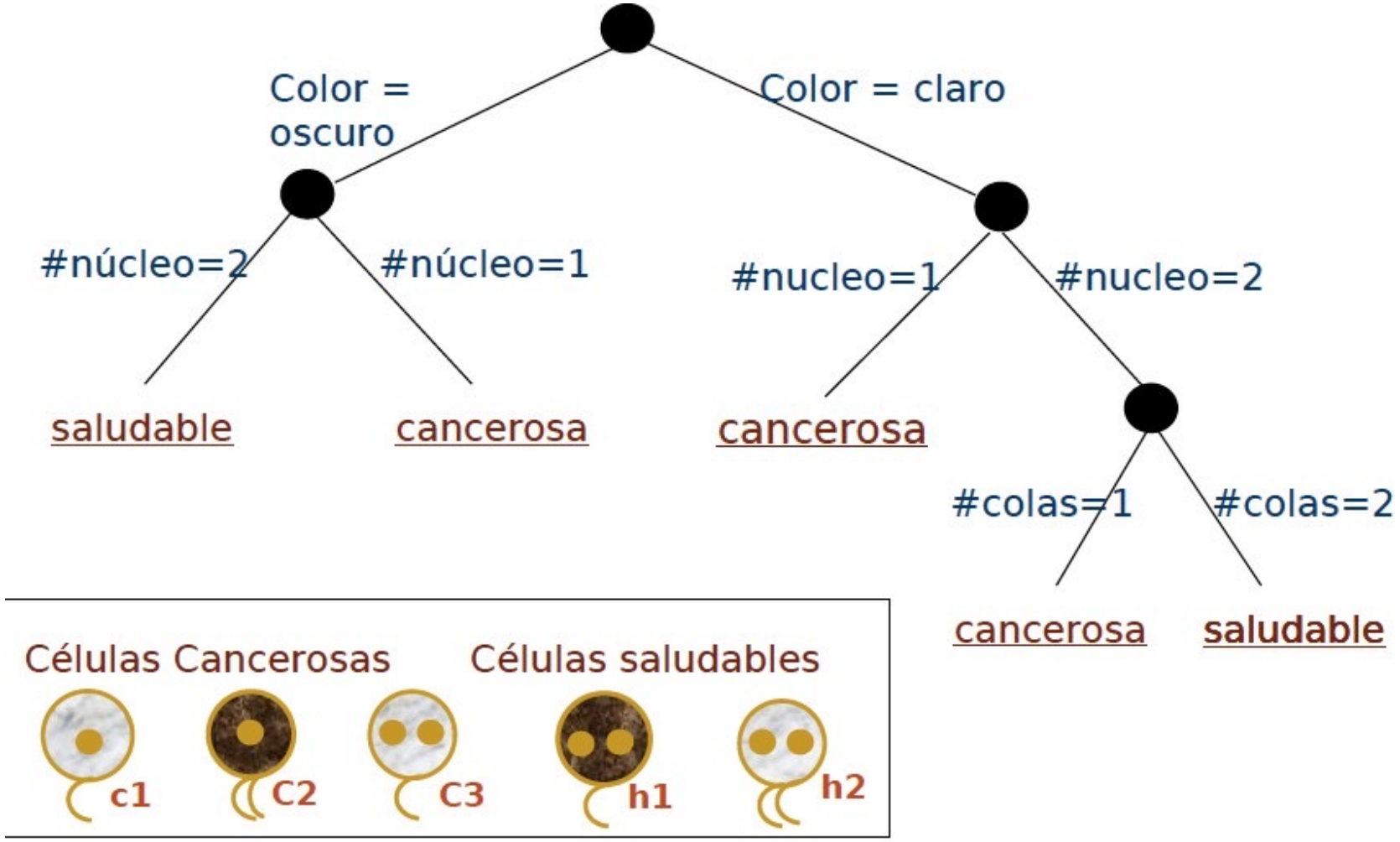


3. Árboles de decisión

Ejemplo 2: ¿Qué factores determinan las células cancerígenas?

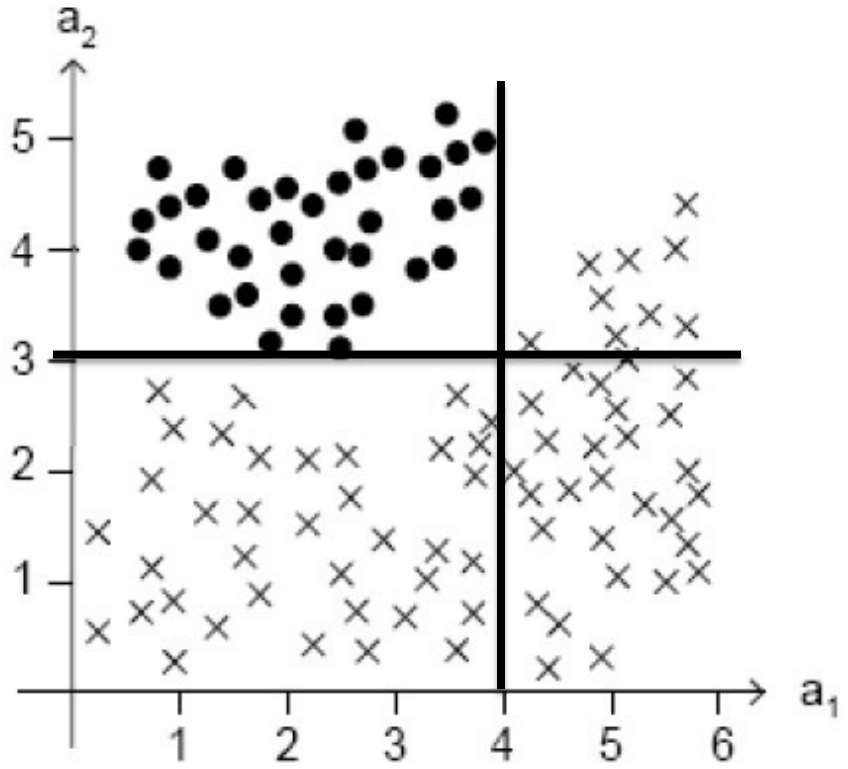


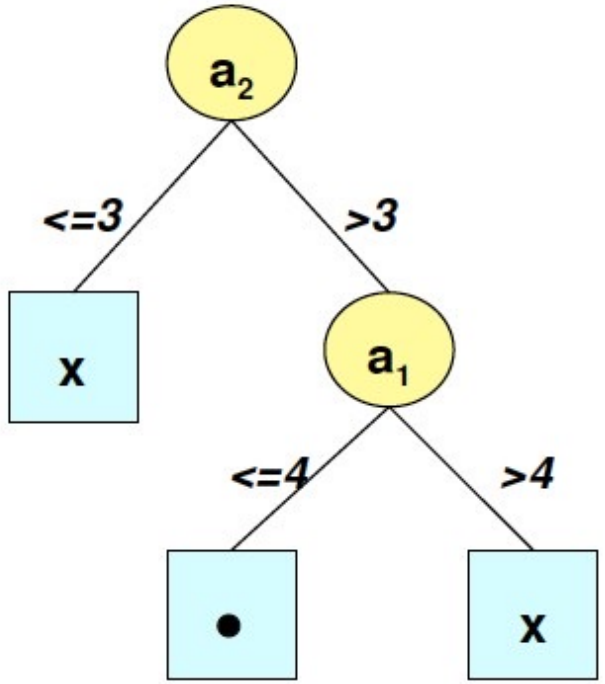
3. Árboles de decisión



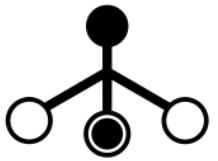
3. Árboles de decisión

Ejemplo 3: Clasificar un punto como: ● ó **x**





3. Árboles de decisión

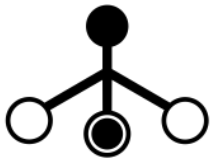


- ¿Cómo se construye un árbol?
  - Hay que encontrar qué atributos son los que mejor dividen al conjunto. Estos atributos serán los nodos del árbol.
- Para encontrar estos atributos:
  - Entropía: mide el (des)orden de los atributos frente a la variable objetivo
  - Ganancia de información: mide cómo de valiosa es la información aportada
  - Gini: índice que mide la dispersión estadística

https://www.analyticsvidhya.com/blog/2016/04/tree-based-algorithms-complete-tutorial-scratch-in-python/

http://www.r2d3.us/una-introduccion-visual-al-machine-learning-1/

3. Árboles de decisión

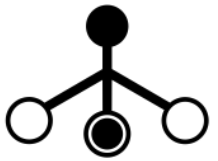


Ventajas

- **La simplicidad del modelo**. Los modelos de árbol de decisión son fácilmente legibles, pues es posible observar de forma jerárquica cómo intervienen los atributos en la toma de decisión de cara a producir una predicción. Además, los atributos situados en la parte superior del árbol poseen mayor relevancia para el modelo que los situados en la parte inferior, lo cual ayuda a entender cuáles son las variables críticas del problema.

- **La fácil interpretabilidad de las predicciones**. Las predicciones llevadas a cabo por un modelo de árbol de decisión pueden ser fácilmente explicadas trazando el camino que lleva desde el nodo raíz de árbol hasta el nodo hoja que produjo la predicción. De esta forma, es posible conocer la causa que produjo la predicción, pues basta con extraer la regla “si…, entonces…” a partir del camino antes mencionado.

3. Árboles de decisión

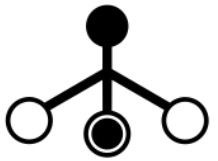


Ventajas

- **El tiempo de ejecución de la predicción muy rápido**. Una vez entrenado un modelo de árbol de decisión, realizar las predicciones es muy rápido, pues tan solo es necesario seguir el camino desde el nodo raíz hacia el nodo hoja al que se llegue tras cumplir las condiciones que satisfaga la instancia de test. Este tiempo es logarítmico con respecto al número de instancias de entrenamiento, con lo cual es muy eficiente.

- **La gran flexibilidad a las características de los datos**. Los algoritmos de árboles de decisión son robustos frente a outliers, valores ausentes y atributos en diferentes escalas de valores. La eficacia del modelo suele verse poco afectada por estos factores. Esto hace que la preparación de datos pueda ser mucho más liviana cuando usamos árboles de decisión.

3. Árboles de decisión

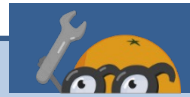


Inconvenientes

- **El riesgo de sobreajuste**. Los algoritmos de árboles de decisión pueden llegar a generar árboles muy grandes, con muchísimos nodos y aristas, que se especializan demasiado en los datos del conjunto de entrenamiento y no generalizan adecuadamente. Este riesgo de sobreajuste aumenta conforme el conjunto de datos posee mayor cantidad de atributos y menor número de instancias. No obstante, existen mecanismos para tratar de evitar el sobreajuste, tales como podar el árbol (eliminar nodos), establecer un umbral mínimo de ejemplos cubiertos por un nodo hoja o limitar la profundidad máxima (número de niveles) del árbol.

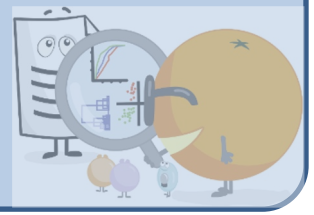
- **La sensibilidad al desbalanceo de clases**. Al igual que la mayoría de los clasificadores no especializados en desbalanceo de clases, la eficacia del modelo queda muy mermada por el hecho de que en el conjunto de datos haya muchas más instancias de una clase que de otras, esto es, si la proporción de ejemplos de cada clase no está equilibrada.

3. Árboles de decisión

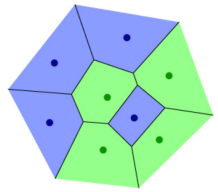


Práctica con Orange

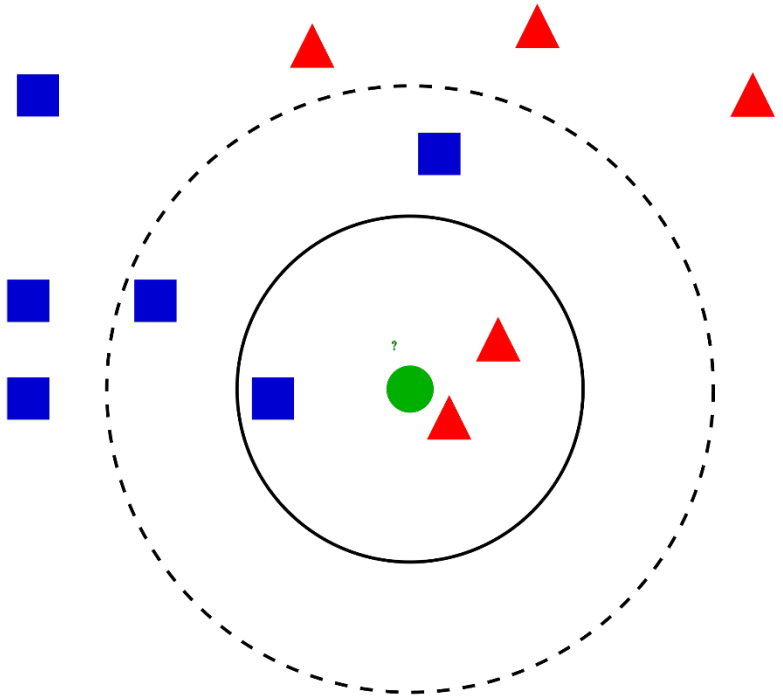
- **Datasets de estudio**
  - Iris
- **Algoritmo**
  - Árbol de decisión
- **Métricas de evaluación**
  - CA
  - Precision
  - Recall
  - F-measure (F1)
- **Validación**
  - Validación cruzada con 10 bolsas



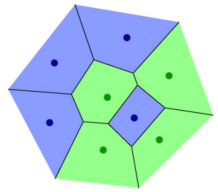
4. Vecinos más cercanos (k-NN)



- Consiste en asignar al punto a predecir la clase mayoritaria entre los k ejemplos más cercanos según una determinada métrica.

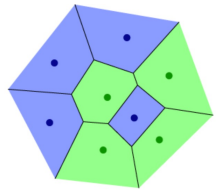


4. Vecinos más cercanos (k-NN)



- No genera un modelo, sino que el conjunto de datos de entrenamiento es el **modelo**.
- No existe fase de **entrenamiento** para crear un modelo, sino que hasta que no llega un dato a clasificar no se pone en marcha el algoritmo de aprendizaje.
- El vecino más cercano no calcula **fronteras de decisión**, aunque son implícitas.

4. Vecinos más cercanos (k-NN)



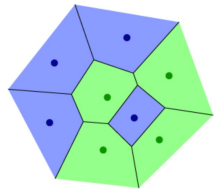
Ejemplo. Predecir el peso de una persona en función de su altura y edad

ID	Height	Age	Weight
1	5	45	77
2	5.11	26	47
3	5.6	30	55
4	5.9	34	59
5	4.8	40	72
6	5.8	36	60
7	5.3	19	40
8	5.8	28	60
9	5.5	23	45
10	5.6	32	58
11	5.5	38	?

Tabla de datos

- 10 personas entrenamiento
- 1 persona test

4. Vecinos más cercanos (k-NN)



Ejemplo. Predecir el peso de una persona en función de su altura y edad

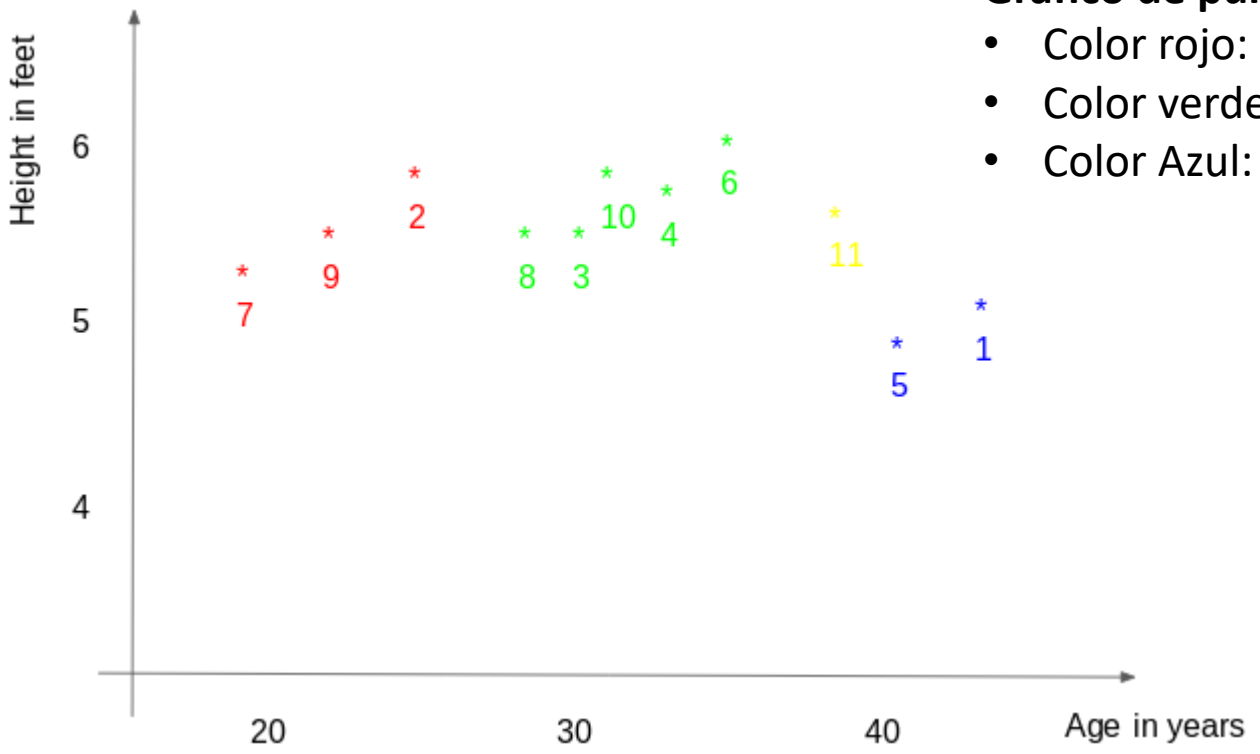
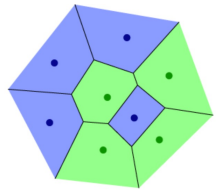


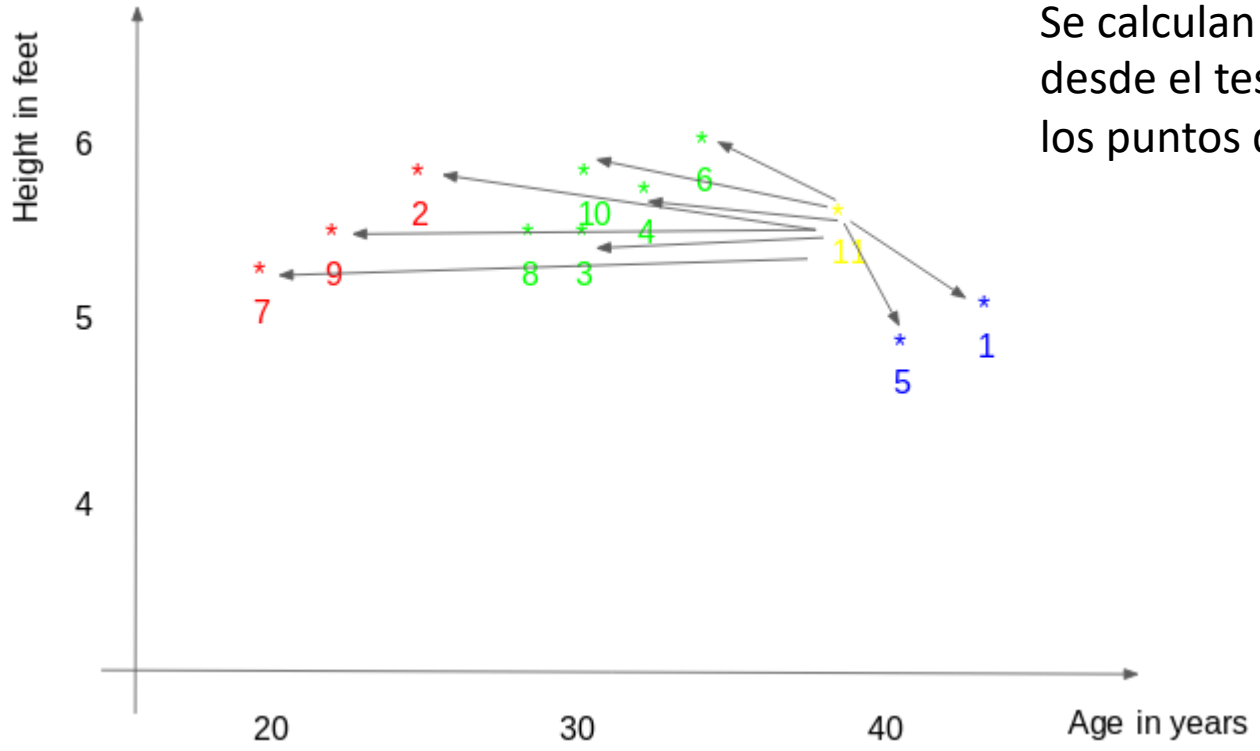
Gráfico de puntos

- Color rojo: pesos bajos
- Color verde pesos medios
- Color Azul: pesos altos

4. Vecinos más cercanos (k-NN)



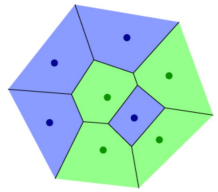
Ejemplo. Predecir el peso de una persona en función de su altura y edad



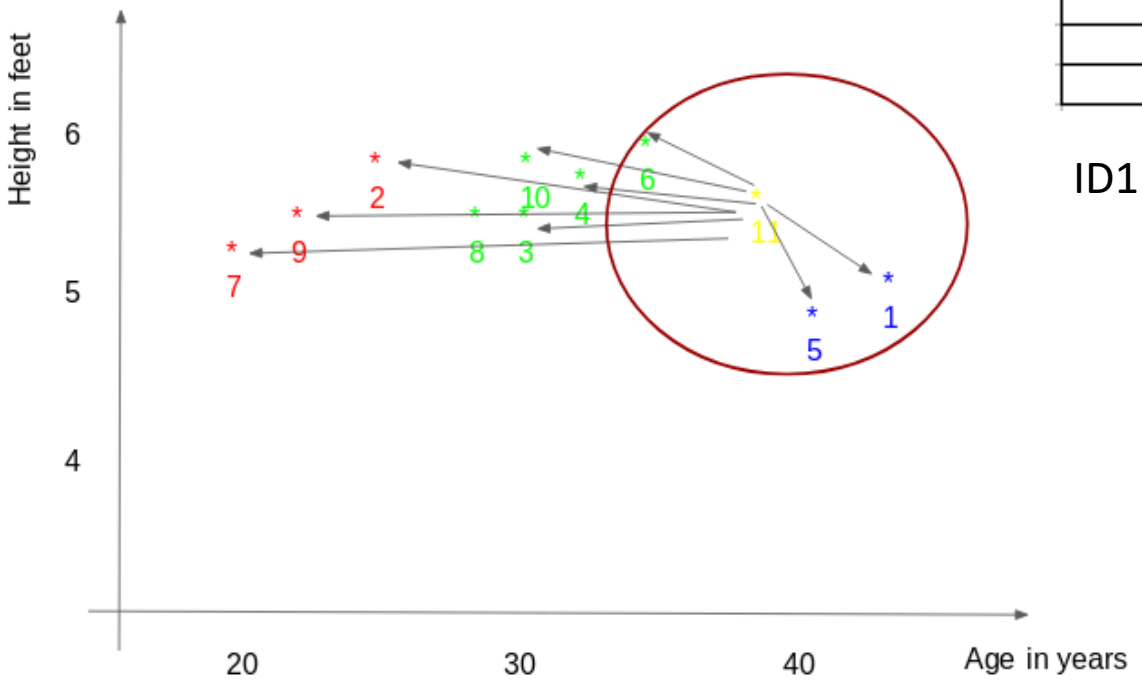
Paso 1

Se calculan las distancias desde el test (11) hacia todos los puntos de entrenamiento.

4. Vecinos más cercanos (k-NN)



Ejemplo. Predecir el peso de una persona en función de su altura y edad



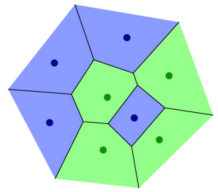
Paso 2

Los tres vecinos más cercanos:

ID	Height	Age	Weight
1	5	45	77
5	4.8	40	72
6	5.8	36	60

ID11 = (77+72+60)/3 = **69.66 kg**

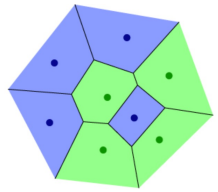
4. Vecinos más cercanos (k-NN)



Ventajas

- **La simplicidad del modelo**. El modelo es tan simple que realmente no tiene modelo propio, tan solo los datos de entrenamiento.
- **La fácil interpretabilidad de las predicciones**. Las predicciones realizadas por el algoritmo KNN pueden ser fácilmente interpretadas si mostramos un resumen de los vecinos utilizados en la predicción.
- **El tiempo de ejecución del entrenamiento prácticamente nulo**. El tiempo empleado en el entrenamiento consiste únicamente en almacenar una referencia en los datos de entrenamiento para el cálculo de distancias durante la fase de predicción. No hay construcción de ningún modelo a partir de los datos.

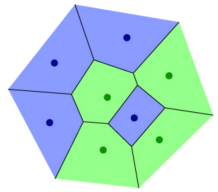
4. Vecinos más cercanos (k-NN)

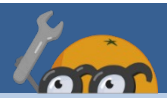


Inconvenientes

- **El tiempo de ejecución alto en la fase de predicción**. Para realizar la predicción a partir de un ejemplo de test, es preciso calcular las distancias a todos los ejemplos de entrenamiento, lo cual es el proceso más costoso del algoritmo. Este puede ser un importante hándicap cuando el conjunto de datos es enorme, con varios millones de instancias.
- **La falta de generalización**. Al no crearse ningún modelo a partir de los datos, si estos poseen anomalías, la eficacia del algoritmo puede verse afectada, especialmente con valores bajos de k. Además, si los datos están muy dispersos en el espacio de atributos o bien no son representativos, los errores de generalización del modelo pueden ser elevados.
- **La necesidad de atributos relevantes y en igual escala**. La presencia de atributos en diferentes escalas puede afectar negativamente a los resultados. Además, los datos con una alta dimensionalidad (gran cantidad de atributos) deberían ser reducidos mediante una selección de los atributos más relevantes.

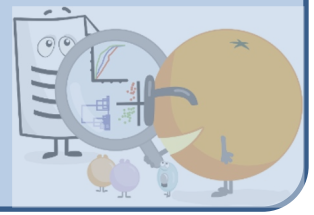
4. Ejercicio: Vecinos más cercanos



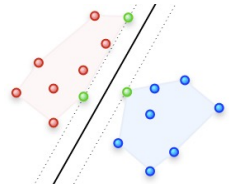


Práctica con Orange

- **Datasets de estudio**
  - Iris
- **Algoritmo**
  - Vecinos más cercanos (k=1)
  - Vecinos más cercanos (k=3)
- **Métricas de evaluación**
  - CA
  - Precision
  - Recall
  - F-measure (F1)
- **Validación**
  - Validación cruzada con 10 bolsas

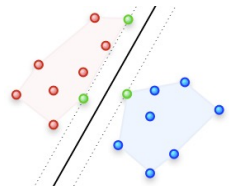


5. Máquinas de vectores soporte

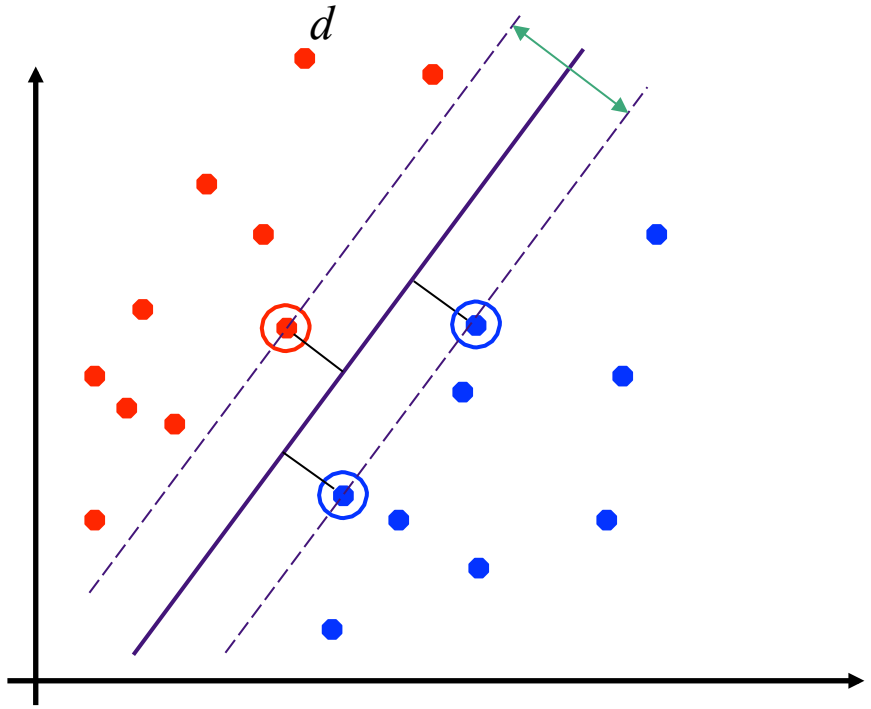


- Introducidas en los 90 por Vapnik para problemas de clasificación
- Calcula una recta o una frontera no lineal resolviendo un problema de optimización
- Para calcular una frontera no lineal hay que usar un kernel

5. Máquinas de vectores soporte

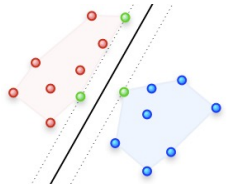


- Puntos más cercanos a la recta: vectores soporte
- El margen es la distancia mínima de vectores soporte al hiperplano
- Objetivo: calcular hiperplano que maximiza el margen



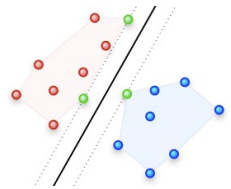
Caso separable linealmente

5. Máquinas de vectores soporte

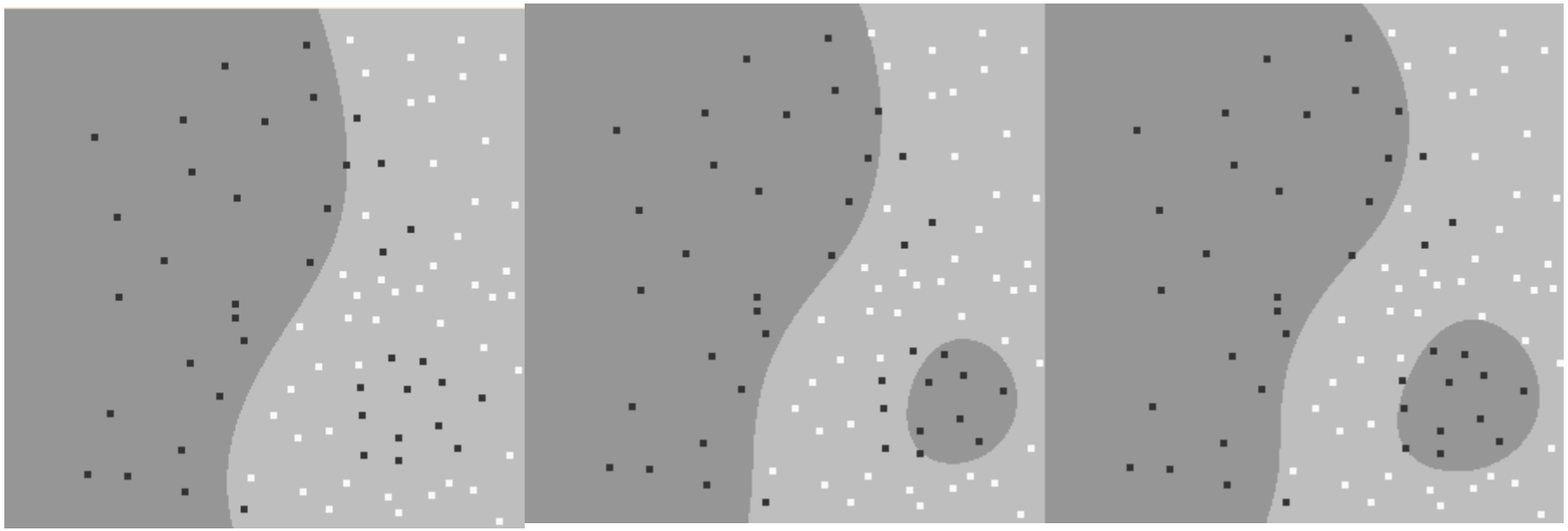


- ¿Y si queremos separadores no lineales?
  - Usar un kernel
- Kernel:
  - Lineal (para separadores lineales)
  - Cuadrático
  - Polinómico de grado d
  - Gaussiano de escala σ

5. Máquinas de vectores soporte

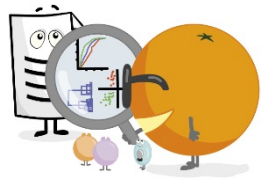


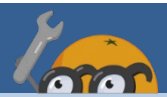
Parámetro C: Parámetro del método de optimización que se usa para calcular los vectores soporte y la frontera. Nos permite regular el compromiso entre coste y precisión (**evitar el sobreajuste**).



más bajo intermedio más alto

6. Ejercicio de clasificación





Práctica con Orange

- **Datasets de estudio**
  - Zoo (Orange)
  - Wine (Orange)
  - Adult (Orange)
- **Algoritmos**
  - Naive Bayes
  - Árbol de decisión
  - KNN
  - Máquinas vectores soporte
- **Métricas de evaluación**
  - CA
  - Precision
  - Recall
  - F-measure (F1)
- **Validación**
  - Validación cruzada con 10 bolsas
- **Análisis**
  - Analizar predicciones
- **Preparar un informe en PDF con todos los resultados obtenidos (herramienta de informes de Orange)**