Aprendizaje supervisado: Clasificación

1. El problema de clasificación.

1. Aplicaciones.

1. El algoritmo K-NN.

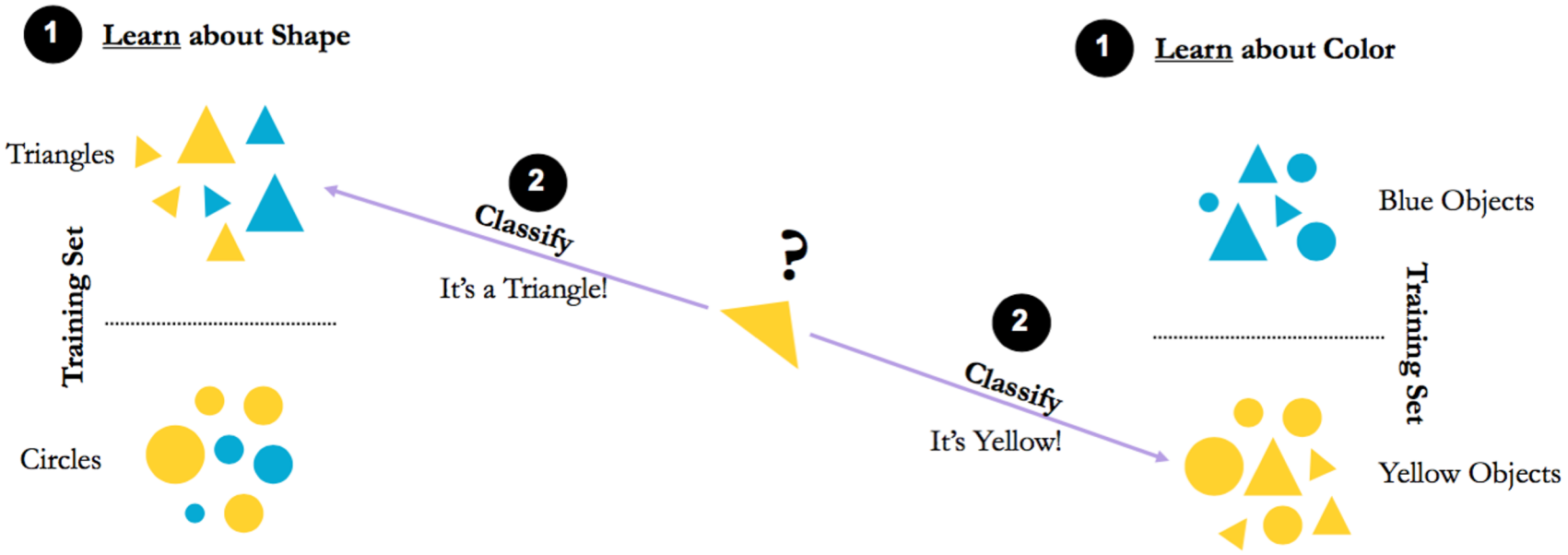
El problema de clasificación

El problema de clasificación

Previously in Machine Learning...

Aprendizaje Supervisado

Un maestro provee conjuntos de entrenamiento etiquetados, usados para *entrenar* un clasificador



La principal diferencia con regresión es que en clasificación la variable dependiente es una variable categórica.

El problema de clasificación

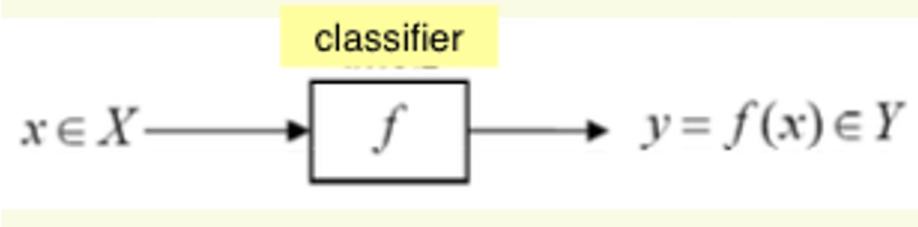
Matemáticamente:

Un clasificador es una función $f : X \longrightarrow Y$

X es el espacio de entrada, y $x \in X$ es dato del espacio de entrada.

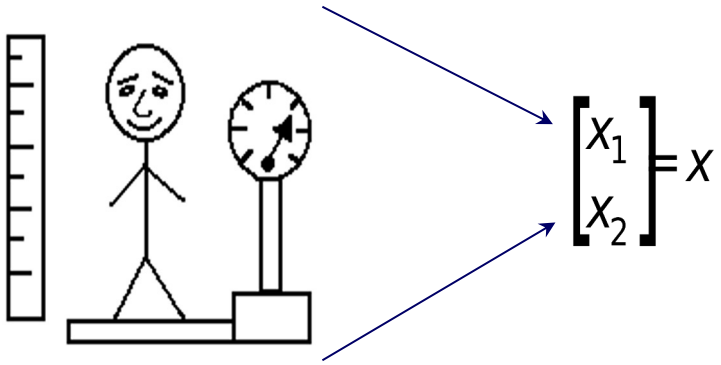
Un dato del espacio de entrada es un x tal que $x = \{x_1, x_2, ..., x_D \mid c_k\} \in \mathbb{R}^{D+1}$ donde D es la dimensión del dato y c_k es la etiqueta (o clase) k del dato.

Ω es un conjunto finito de etiquetas tal que $\Omega = \{c_1, ..., c_k, ..., c_K\}$ y K es el número de etiquetas distintas existentes en el conjunto finito Ω .



El problema de clasificación

EJEMPLO:



Espacio de entrada: $X = \Re^2$

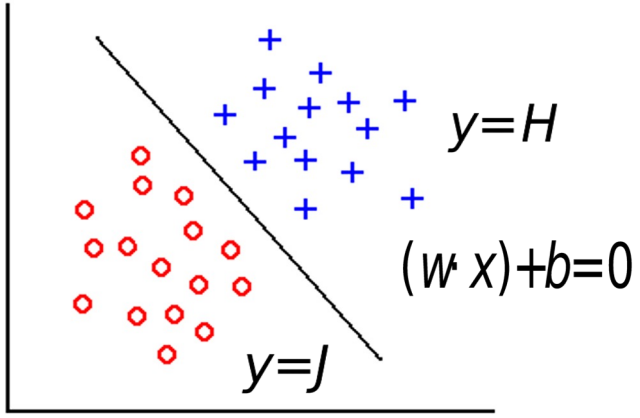
Espacio de salida: $Y = \{H, J\}$

Datos de entrenamiento

$$\{(x_1, y_1), \ldots, (x_l, y_l)\}$$

Clasificador Lineal:

$$q(x) = \begin{cases} H & \text{if } (w \cdot x) + b \geq 0 \\ J & \text{if } (w \cdot x) + b < 0 \end{cases}$$



El problema de clasificación

Se debe cumplir, al igual que en regresión, la división del data set original en training set y testing set.

También debemos evaluar los algoritmos para ver que tan bien, o mal, lo están haciendo. Para eso veremos, las métricas usadas para esto.

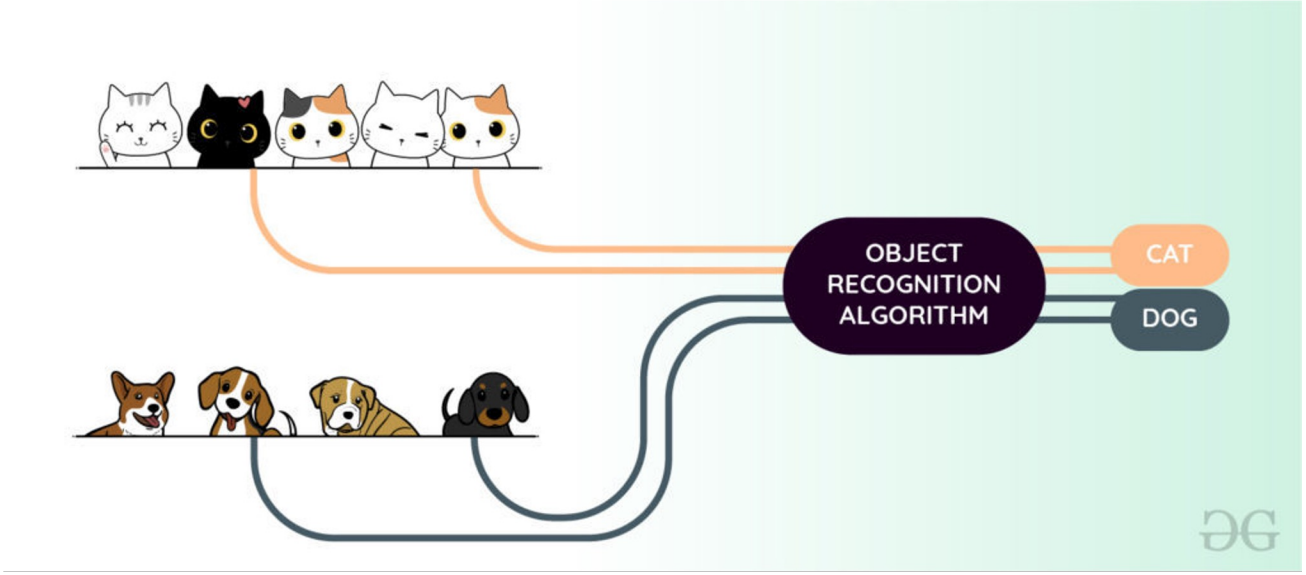
Aplicaciones de clasificación

Aplicaciones

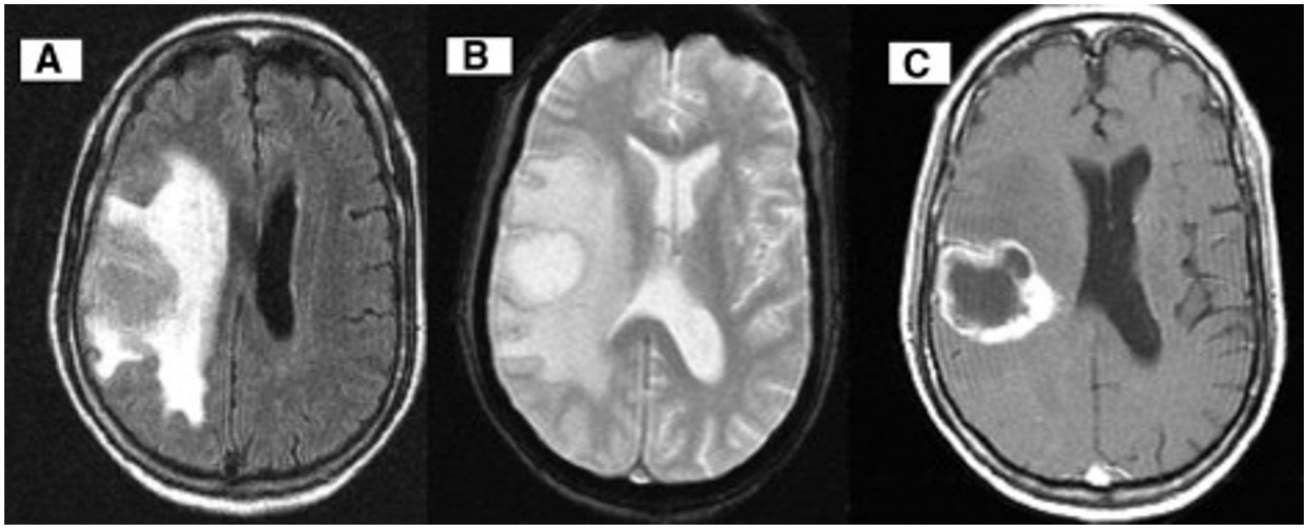
Fuente: https://searchsecurity.techtarget.com/definition/biometric-authentication

Aplicaciones

Images recognition



Tumor recognition



K-NN como algoritmo de clasificación

K-NN

K-NN: K- Nearest Neighbors

- El clasificador K-NN está basado en la noción de vecinos más cercanos (Nearest Neighbour), por lo tanto, se dice que está basado en instancias (ejemplos).

- En el área de los clasificadores, K-NN es especial, ya que no construye un modelo interno para poder predecir clases de datos futuros.

- No aprende de los datos, sino que los guarda, junto con información adicional para que luego pueda decidir a qué clase pertenece el dato de test.

K-NN

- K-NN es muy usado por su fácil implementación e idea intuitiva de cómo trabaja.

- Otra cualidad es que su complejidad computacional es baja (comparada con clasificadores más tradicionales).

- Trabaja fácilmente con problemas de 3 o más clases.

- Lo malo:
  - Muy dependiente de sus parámetros.
  - Algunos de sus parámetros son altamente dependientes de los datos.
  - Problema en ambos casos: muchas dimensiones y muchos ejemplos (aunque hay soluciones a este último).

K-NN

- K-NN debe construir una matriz de distancia.

		X_1	...	X_j	...	X_n	C
$(\mathbf{x}_1, c_1)$	1	x_{11}	...	x_{1j}	...	x_{1n}	c_1
	$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$
$(\mathbf{x}_i, c_i)$	i	x_{i1}	...	x_{ij}	...	x_{in}	c_i
	$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$
$(\mathbf{x}_N, c_N)$	N	x_{N1}	...	x_{Nj}	...	x_{Nn}	c_N
Dato de test → $\mathbf{x}$	$N+1$	$x_{N+1,1}$	...	$x_{N+1,j}$	...	$X_{N+1,n}$	?

- Se debe elegir el número de vecinos a considerar:
  - K vecinos.
  - Radio para la construcción de la hiper-esfera.

- Se debe elegir la distancia a utilizar.

K-NN

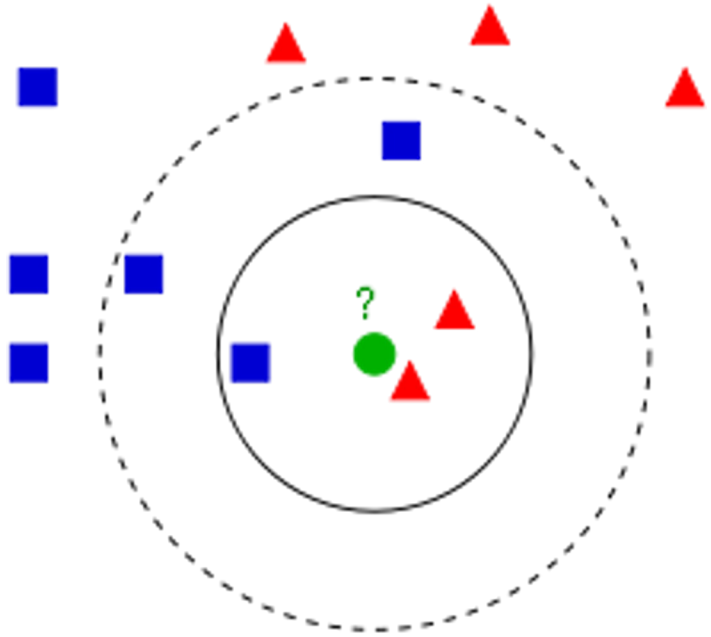
- Las distancias pueden ser:
  - Euclidiana
  - Mahalanobis
  - Minkowski
  - Jaccard
  - Correlation
  - Etc.

- La distancia dependerá de los tipos de datos (numéricos, categóricos, etc.) y de su distribución.

- Cuando la dimensionalidad es muy grande, el concepto de distancia pierde significado.

K-NN

Idea gráfica:



Ejemplo con radio r de 2 valores.

K-NN

Una vez la métrica de distancia está elegida, es necesario elegir la estrategia de cómo encontrar los vecinos más cercanos.

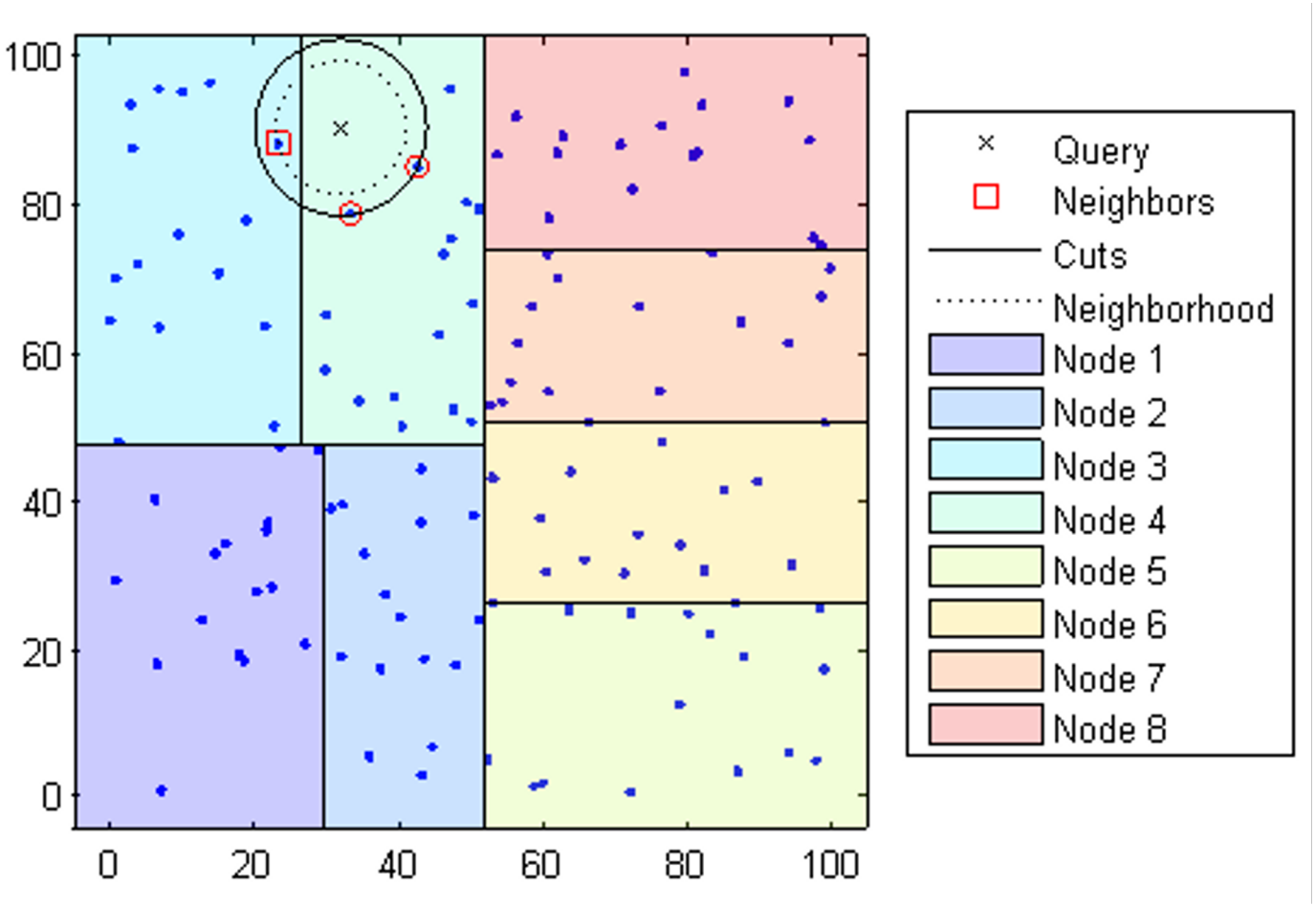
Existe 2 estrategias las que son más usadas:

- **Brute Force:** Se calcula la distancia del dato de test a todos los demás datos de training. Se eligen los k datos más cercanos al dato de testing.

- **Kd.Tree:** divide el espacio de los datos en regiones, según densidad de los datos. Luego selecciona los ejemplos más cercanos dentro de la división que pertenece el dato de test. Con esta distancia, crea una hiper-esfera que busca por más vecinos.

K-NN

Ejemplo de Kd.Tree:



K-NN

Finalmente, mediante votación, se define la clase del dato de test.

Existe diversas variantes:

- **K-NN con rechazo:** una garantía mínima (umbral) para tomar la decisión

- **K-NN con distancia media:** se calcula la distancia media de los ejemplos pertenecientes a cada clase.

- **K-NN con ponderación en ejemplos y características:** cada ejemplo y su similaridad es ponderada según la inversa de la distancia y/o según las características.

K-NN

Ejemplo con el dataset Iris: