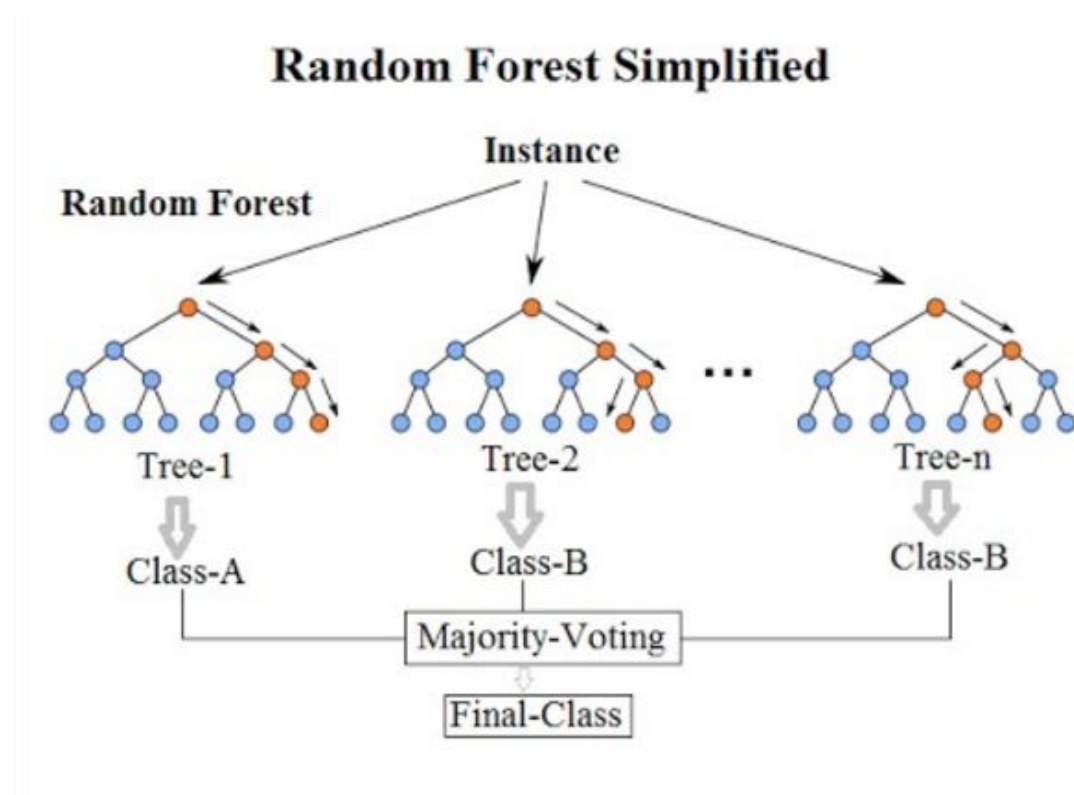


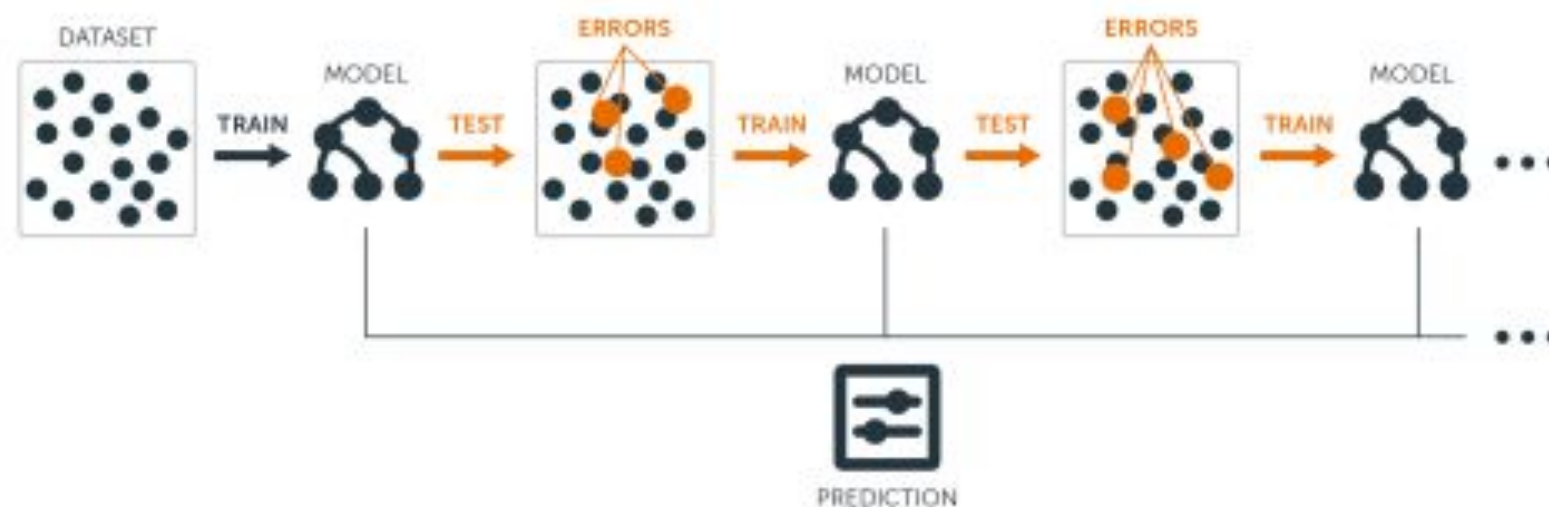
Random Forest

- Es posible combinar varios árboles de decisión para tener modelos con mayor capacidad predictiva.
- Random Forest:
 - un **ensamble** de árboles de decisión (a veces de un sólo nivel).
 - Cada árbol se entrena con distintas muestras aleatorias (con reemplazo) del dataset de entrenamiento.
 - Luego los distintos árboles votan para hacer predicciones finales



Gradient Boosting

- Boosting es otro enfoque para combinar varios clasificadores
- Idea: se entrenan modelos de manera secuencial donde cada modelo siguiente trata de corregir los errores del modelo anterior
- En Gradient Boosting los modelos son **árboles de decisión**.
- Cada árbol se entrena para predecir los errores del anterior.
- La predicción final es una suma ponderada de todos los árboles.
- Existen implementaciones eficientes de la idea como XGBoost y CatBoost para atributos categóricos.

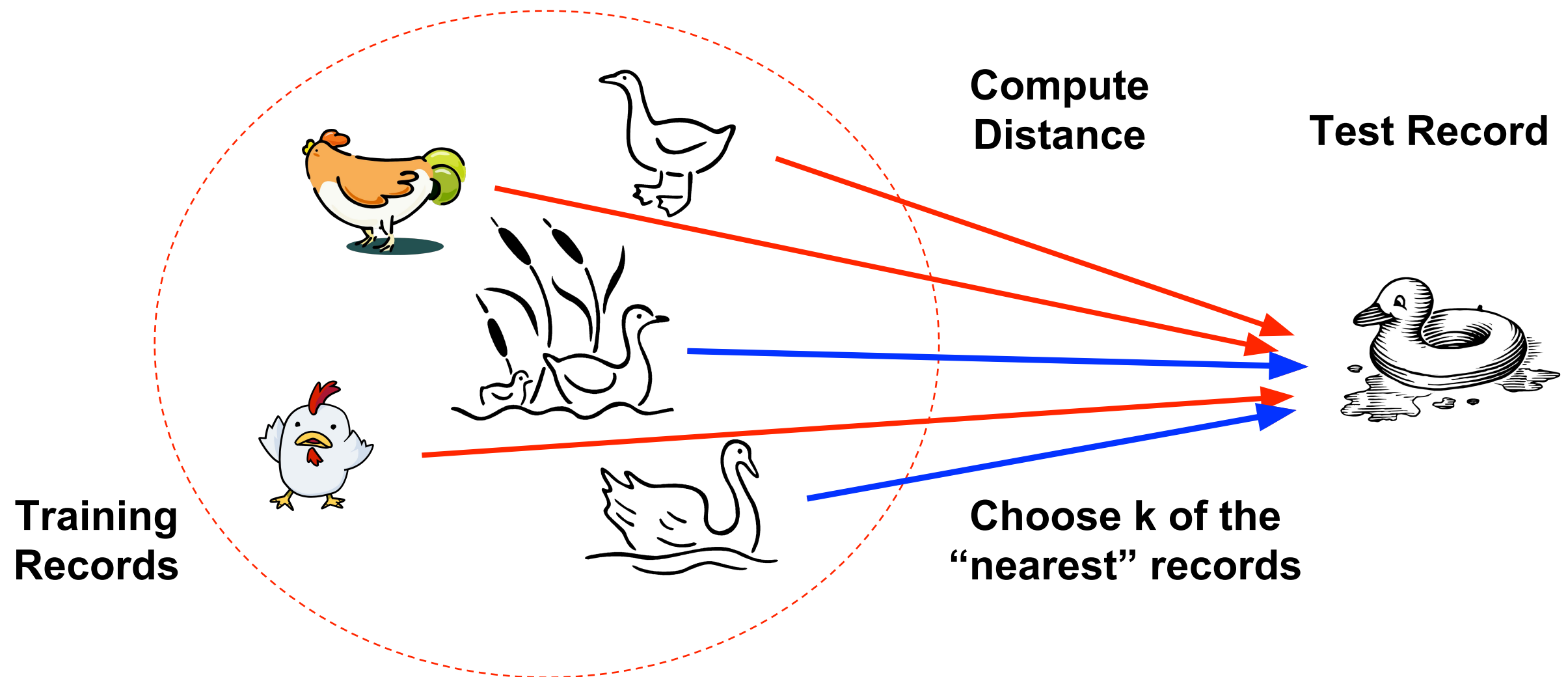


Clasificador KNN

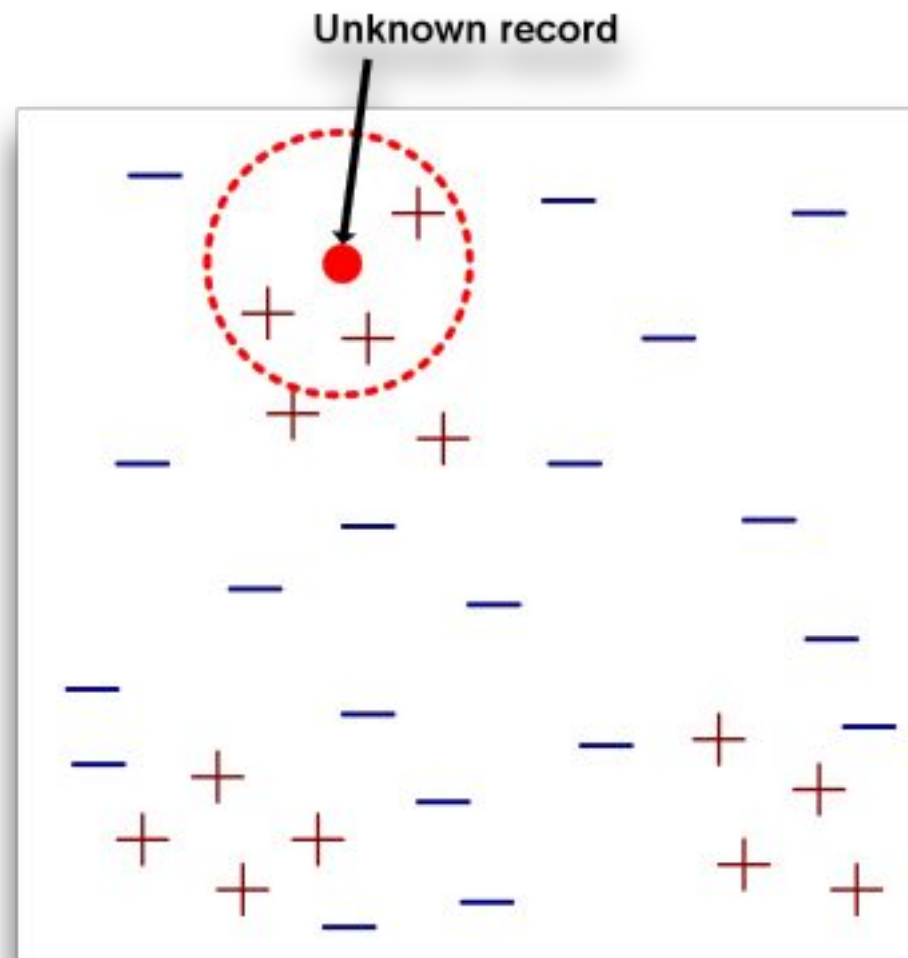
- Nearest Neighbor Classifier (o k-nn)
- Es un clasificador basado en **instancias**
- Conocido como **lazy**
 - Usa los **k** puntos más cercanos (nearest neighbors) para realizar la clasificación

Clasificadores KNN

- Idea:
 - **If it walks like a duck, quacks like a duck, then it's probably a duck**



Clasificadores KNN



- Necesita 3 cosas
 - Set de records almacenados.
 - Métrica de distancia para calcular la distancia entre records.
 - Valor de k , el número de vecinos cercanos a obtener.
- Para clasificar un récord nuevo
 - Calcular la distancia los los récords almacenados.
 - Identificar k nearest neighbors .
 - Utilizar la clase de los knn para asignar la clase al record nuevo (e.j. voto de la mayoría).

Métricas de distancia

- Para atributos numéricos usamos la distancia euclidiana:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{k=1}^n (x_k - y_k)^2},$$

- Una versión más general es la distancia de **Minkowsky** (r=1 => distancia Manhattan, r=2 => distancia euclideana)

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{k=1}^n |x_k - y_k|^r \right)^{1/r}$$

- Es muy importante que los atributos estén normalizados.

Escalando atributos

- Problemas de escalas

- Atributos deben ser escalados para prevenir que algún atributo domine la métrica de distancia

- Ejemplos:

- La altura de una persona puede variar entre 1.5m a 1.8m

- El peso puede variar entre 40 kg a 150 kg

- El ingreso de una persona puede variar entre \$150K a \$10M

Técnicas para escalar atributos

- Normalización a media cero y varianza unitaria:

$$\frac{x - \mu_x}{\sigma_x}$$

- Normalización a rango entre 0 y 1:

$$\frac{x - \min_x}{\max_x - \min_x}$$

OJO: Apliquen la misma transformación a los datos de training y testing -> los valores de normalización se calculan sobre los datos de training.

Distancia y similitud

- Es importante distinguir entre métricas de similitud y métricas de distancia.
- Similitud: entre más cerca dos objetos mayor el valor de la métrica.
- Distancia: entre más lejos dos objetos mayor el valor de la métrica.

Attribute Type	Dissimilarity	Similarity
Nominal	$d = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{if } x \neq y \end{cases}$	$s = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$
Ordinal	$d = x - y / (n - 1)$ (values mapped to integers 0 to $n-1$, where n is the number of values)	$s = 1 - d$
Interval or Ratio	$d = x - y $	$s = -d, s = \frac{1}{1+d}, s = e^{-d},$ $s = 1 - \frac{d - \min_d}{\max_d - \min_d}$

Similitud Coseno

- Cuando nuestros objetos son vectores sparse (muchas columnas con cero) es conveniente usar la similitud coseno.
- Corresponde al coseno del ángulo entre los dos vectores.

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|} \quad \|\mathbf{x}\| = \sqrt{\sum_{k=1}^n x_k^2} = \sqrt{\mathbf{x} \cdot \mathbf{x}}.$$

- Un ejemplo común es cuando tratamos documentos como bolsas de palabras (cada columna es una palabra del vocabulario).

Ejemplo:

$$\mathbf{x} = (3, 2, 0, 5, 0, 0, 0, 2, 0, 0)$$

$$\mathbf{y} = (1, 0, 0, 0, 0, 0, 0, 1, 0, 2)$$

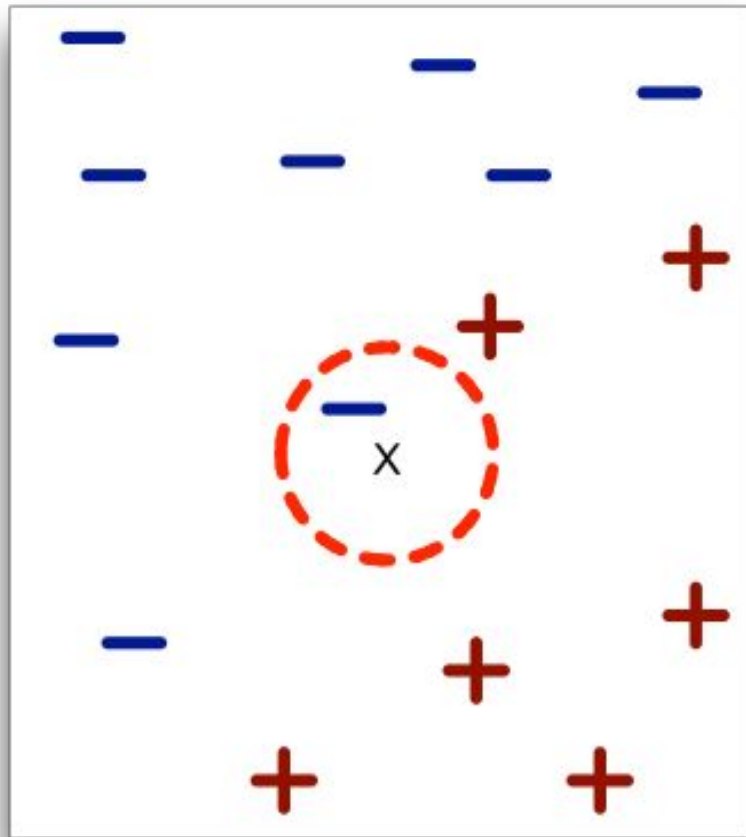
$$\mathbf{x} \cdot \mathbf{y} = 3 * 1 + 2 * 0 + 0 * 0 + 5 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 2 * 1 + 0 * 0 + 0 * 2 = 5$$

$$\|\mathbf{x}\| = \sqrt{3 * 3 + 2 * 2 + 0 * 0 + 5 * 5 + 0 * 0 + 0 * 0 + 0 * 0 + 2 * 2 + 0 * 0 + 0 * 0} = 6.48$$

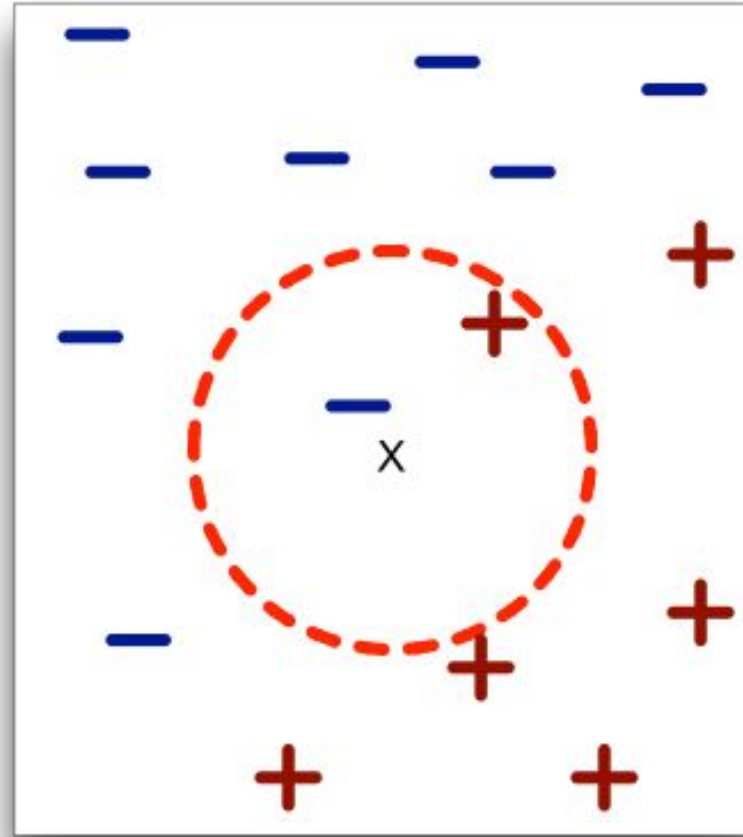
$$\|\mathbf{y}\| = \sqrt{1 * 1 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 1 * 1 + 0 * 0 + 2 * 2} = 2.24$$

$$\cos(\mathbf{x}, \mathbf{y}) = \mathbf{0.31}$$

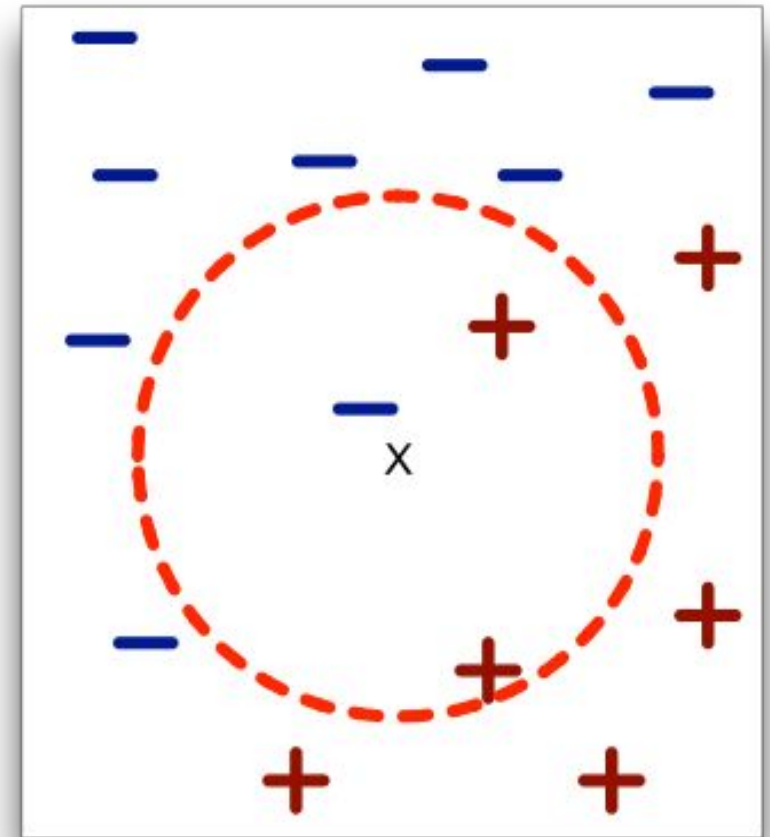
Definición de NN



(a) 1-nearest neighbor



(b) 2-nearest neighbor

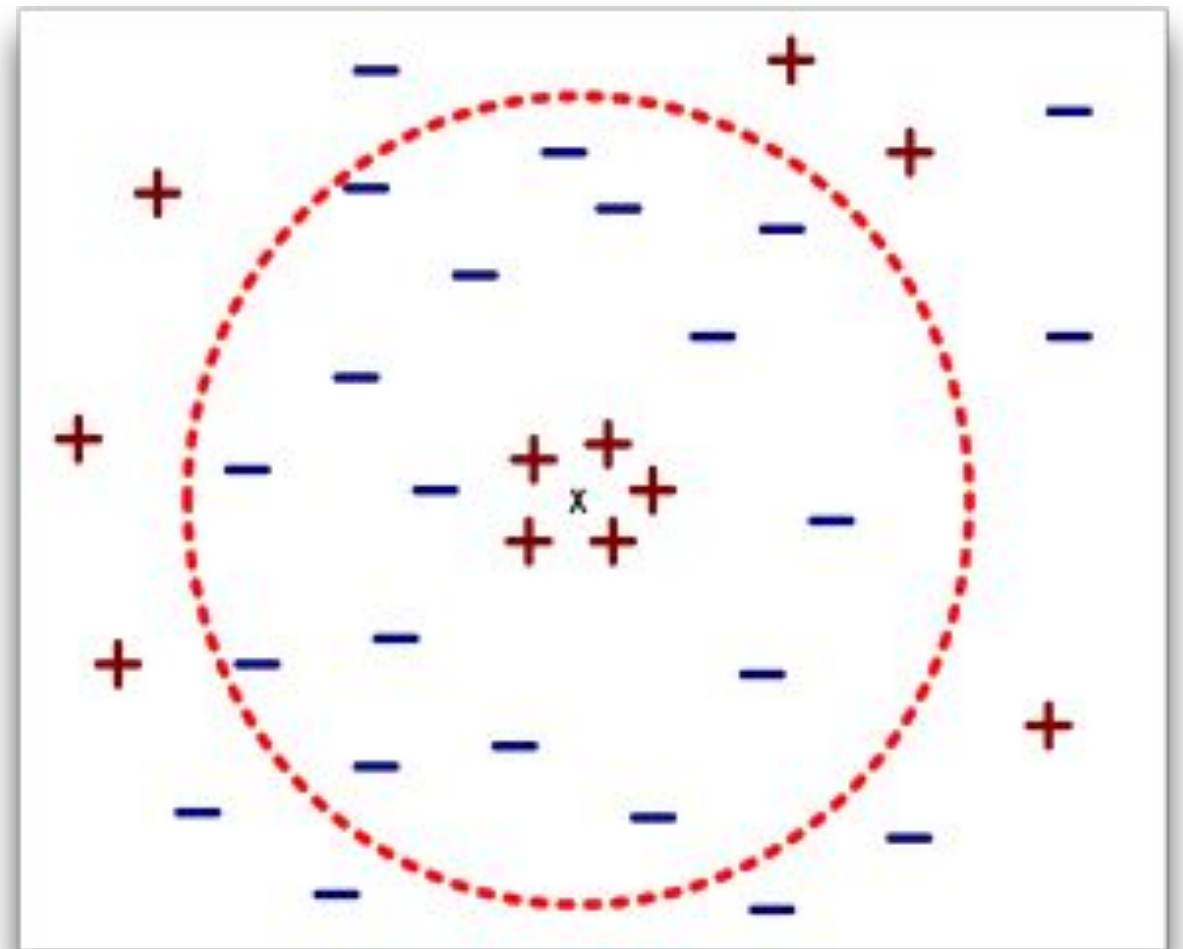


(c) 3-nearest neighbor

K-NN de un record x son los puntos que tienen las k menores distancias a x

Eligiendo el valor de K

- k muy pequeño es susceptible a ruido
- k muy grande puede incluir puntos de otra clase



Clasificación kNN

- Los clasificadores k-NN son **lazy learners**.
- No construyen modelos explícitos, es más flexible ya que no necesita comprometerse con un modelo global a priori.
- Al contrario de otros **eager learners** como los árboles de decisión o clasificadores basados en reglas.
- Es independiente del nro. de clases.
- La clasificación es más costosa (memoria y tiempo).