

# **Aprendizaje Supervisado: Clasificación usando K-Vecinos Cercanos.**

**Guillermo Santamaría-Bonfil**

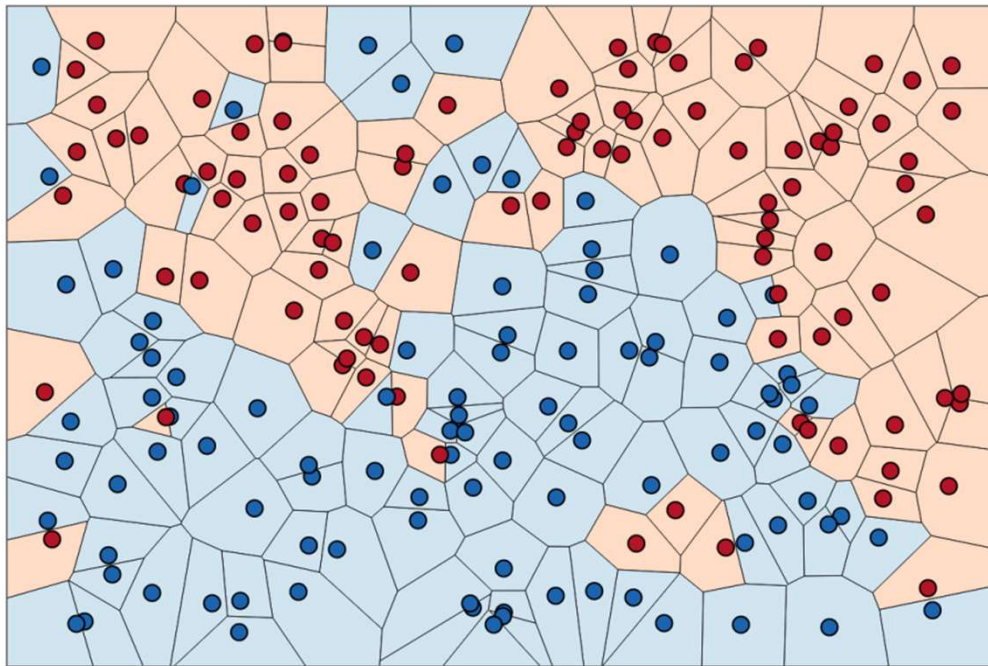
# Agenda

- **K-Vecinos Cercanos.**
  - Teoría
    - # de Vecinos
    - Funciones de distancia
  - Algoritmo



# K-Vecinos Cercanos KNN)

- Surge en 1951 (Fix & Hodges, 1951)
- Técnica no-paramétrica
  - Excepto K
- Almacena todos los posibles casos, y clasifica nuevos casos basados en una medida de similitud.
  - *“Aprendizaje Flojo”*



- Otros nombres:
  - Razonamiento basado en casos.
  - Aprendizaje basado en instancias.
- Suposiciones mínimas sobre los datos.
  - Aproximación local.
- Predicciones precisas pero inestables.

# KNN a grandes rasgos

Existen Varios métodos de NN

## NN

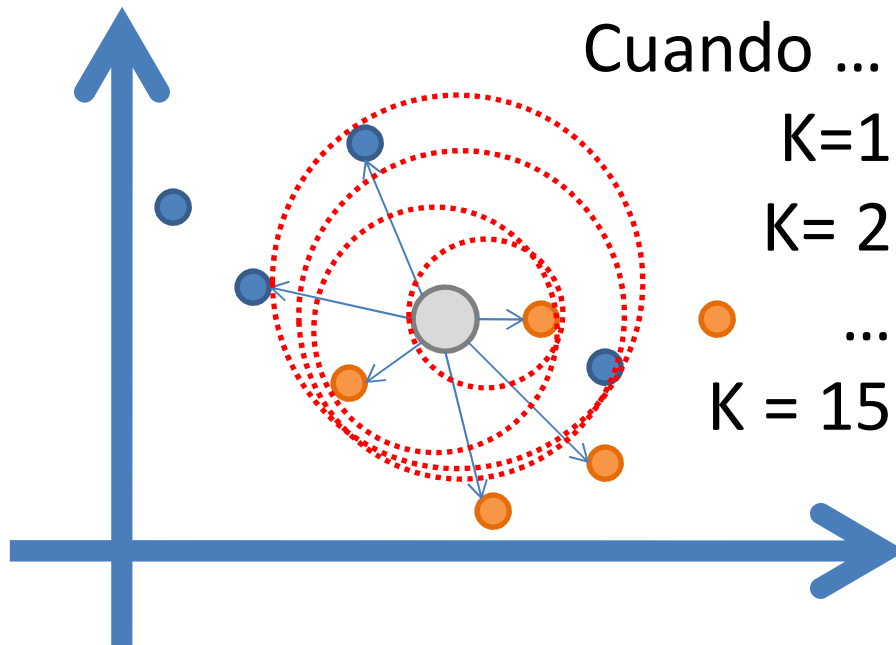
Dado  $\mathbf{x}_{new}$  su clase es  $y_i^*$   $\rightarrow$   
 $\operatorname{argmin}_{\mathbf{x}_i \in X} D(\mathbf{x}_{new}, \mathbf{x}_i), \exists (\mathbf{x}_i^*, y_i^*) \in X \times Y$

## K-NN

Dado  $\mathbf{x}_{new}$  su clase es  
 $\operatorname{majority}(y_i), \forall \mathbf{x}_i \in K$  donde K indica  
el número de vecinos cercanos.

## WK-NN (Weighted)

Dado  $\mathbf{x}_{new}$  su clase es  
 $\operatorname{majority}(\boldsymbol{\beta} y_i), \forall \mathbf{x}_i \in K$  donde K  
indica el número de vecinos cercanos, y  
 $\boldsymbol{\beta}$  es un vector de pesos.



$$D(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_q^N (x_{iq} - x_{jq})^2}$$

# El algoritmo de clasificación usando KNN

Dado  $\mathbf{x}_{new}$  podemos determinar su clase usando el siguiente algoritmo:

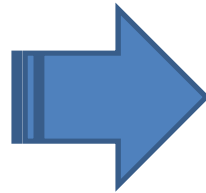
1. Especificar el tamaño de vecindario  $\underline{k}$  ( $\mathbb{Z}_+$ ).
2. Usando una  $D(\mathbf{x}_{new}, \mathbf{x}_j)$  seleccionar  $k$  vecinos más cercanos.
3. Usando la función *mayoria*( $\cdot$ ) determinar la moda entre los  $k$ -nn.
4. Asignar la clase más popular a  $\mathbf{x}_{new}$ .

# Clasificación binaria vía KNN

Dado un conjunto

$$(\bar{x}_i, y_i) \in \mathfrak{R}^N \times \{\pm 1\}$$

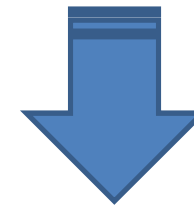
Y un nuevo elemento  $\mathbf{x}_{new}$



$$f(\bar{x})$$

$$\hat{y} = \text{majority}(y_i)_{\mathbf{x}_i \in N_k}$$

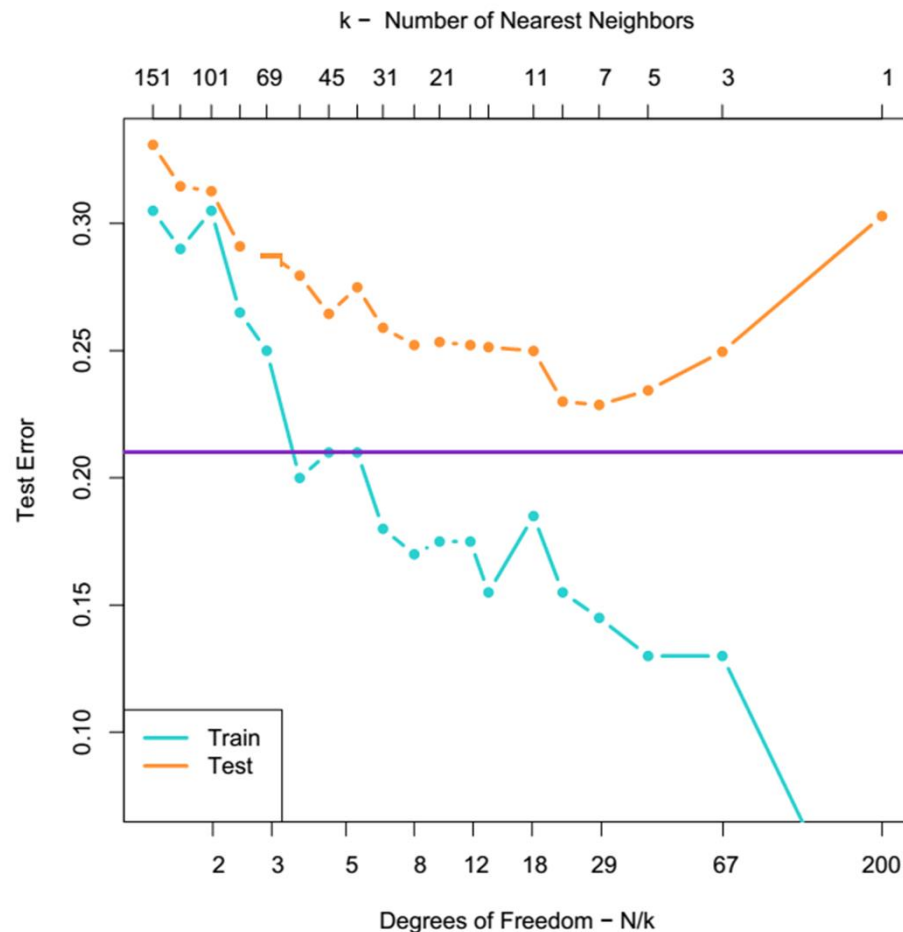
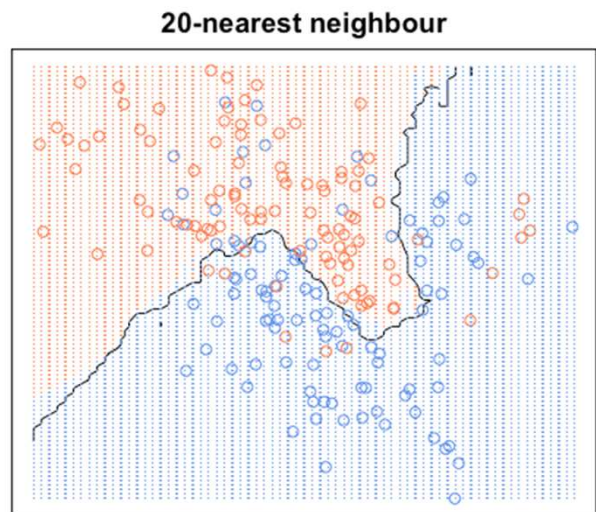
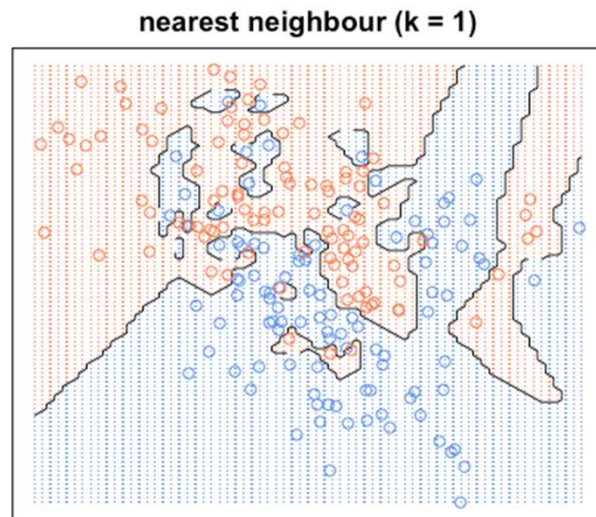
$$D(\mathbf{x}_{new}, \mathbf{x}_i) | \mathbf{x}_i \in N_k$$



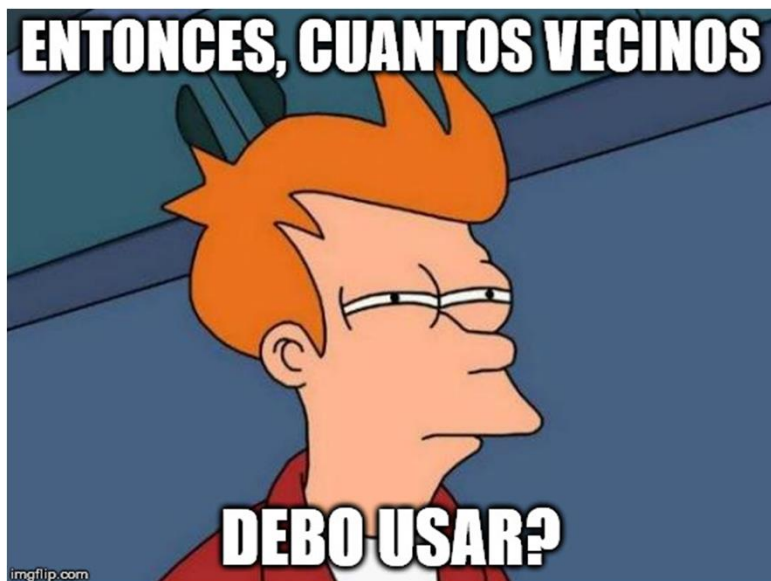
$$\hat{y}_{new}$$

$N_k$ : es el vecindario de  $\mathbf{x}_{new}$  con sus K vecinos más cercanos.

# Más acerca del número de Vecinos: K



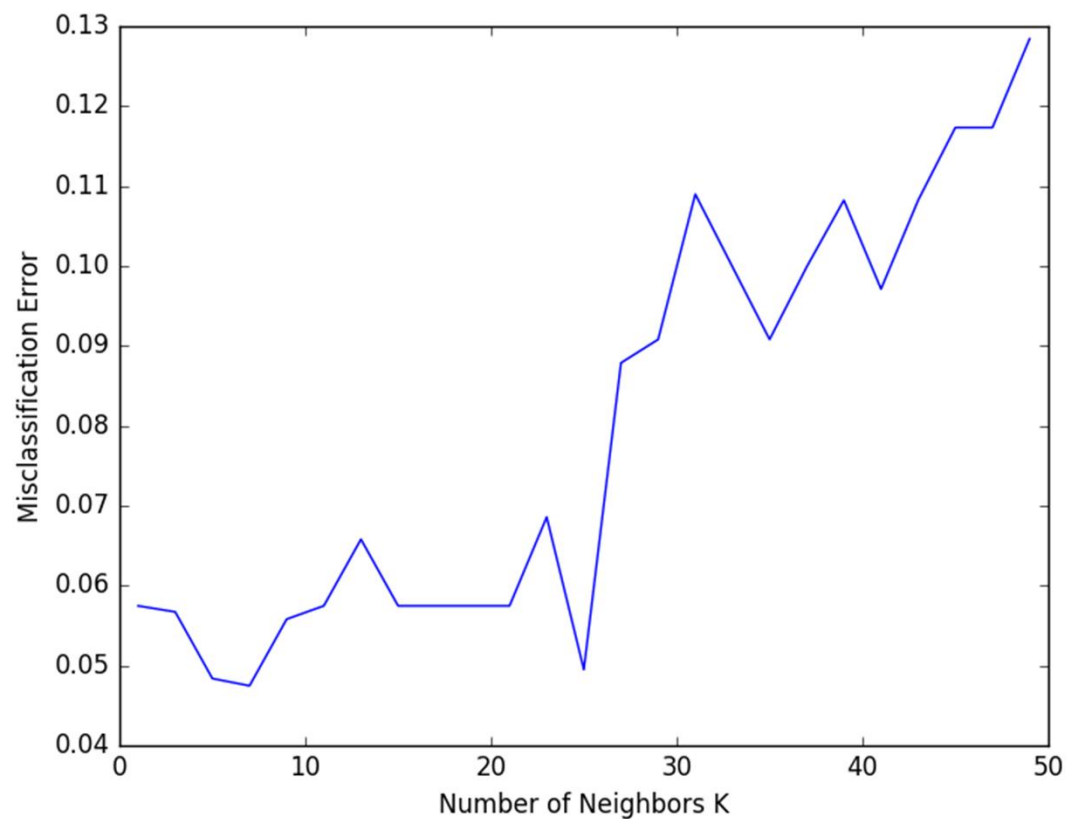




Reglas de dedo:

- $\sqrt{N}$  donde  $N$  es el número de elementos en la muestra
- $K=3, K=5, K=7, \dots?!?!?$

La mejor forma es a través de validación cruzada.





# Funciones de Distancia

- Las funciones de distancia  $D$  funcionan con
  - Variables Nominales/Categóricas
  - Variables Continuas
  - Valores Faltantes
- ¿La clave?
  - Valores pequeños para objetos similares y viceversa.

$$D(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} \approx 0 & \mathbf{x}_i \sim \mathbf{x}_j \\ \infty & \mathbf{x}_i \neq \mathbf{x}_j \end{cases}$$

# Propiedades de las Funciones de Distancia

$$D(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} \approx 0 & \mathbf{x}_i \sim \mathbf{x}_j \\ \infty & \mathbf{x}_i \neq \mathbf{x}_j \end{cases}$$

- $D(\mathbf{x}_i, \mathbf{x}_j) \geq 0$  No-negatividad
- $D(\mathbf{x}_i, \mathbf{x}_i) = 0 \Leftrightarrow \mathbf{x}_i = \mathbf{x}_j$  Identidad
- $D(\mathbf{x}_i, \mathbf{x}_j) = D(\mathbf{x}_j, \mathbf{x}_i)$  Simetría
- $D(\mathbf{x}_i, \mathbf{x}_j) + D(\mathbf{x}_j, \mathbf{x}_k) \geq D(\mathbf{x}_i, \mathbf{x}_k)$  Subaditividad

# Funciones de Distancia

$$D(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{q=1}^N (x_{iq} - x_{jq})^2}$$

Distancia Euclidiana

$$D(\mathbf{x}_i, \mathbf{x}_j) = \sum_{q=1}^N |x_{iq} - x_{jq}|$$

Distancia Manhattan\*

$$D(\mathbf{x}_i, \mathbf{x}_j) = \left( \sum_{q=1}^N (x_{iq} - x_{jq})^q \right)^{1/q}$$

$$D_H = \sum_{q=1}^N |x_i - x_j|$$

$$x_i = x_j \Rightarrow 0$$

$$x_i \neq x_j \Rightarrow 1$$



Distancia de Hamming

Pregunta: ¿A que tipo de datos podemos aplicar esta distancia?

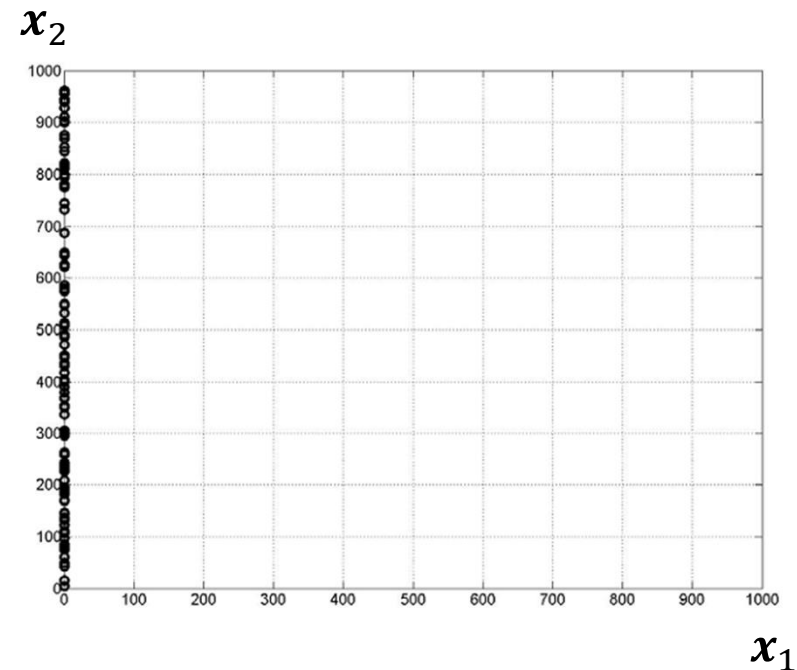
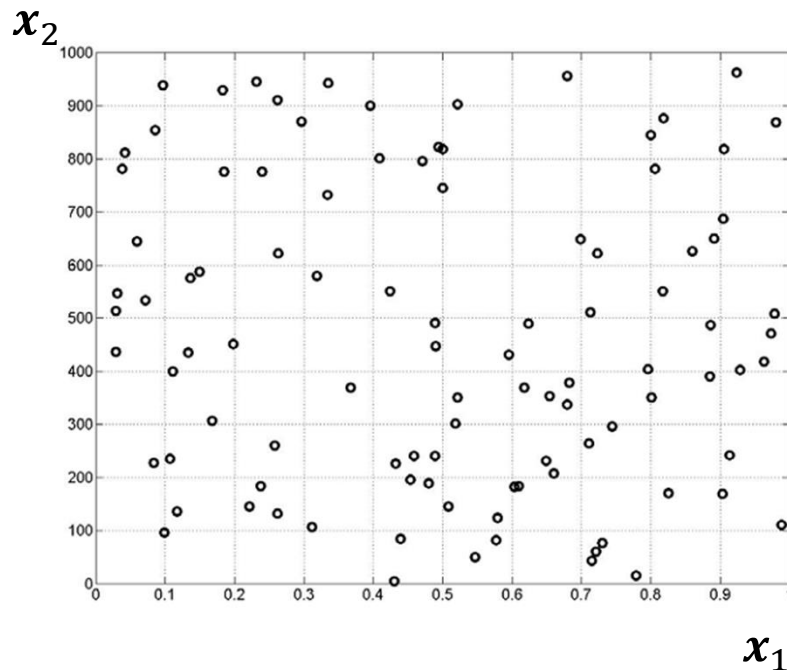
\*La suma total de la diferencia entre las coordenadas x y y.

# Distancias para Variables Continuas

¿Qué pasaría si las variables son medidas en diferentes escalas?

- $x_1 \in [0,1]$
- $x_2 \in [0,1K]$

¿Podemos anticipar un problema con  $D_E$ ?



# Distancias para Variables Continuas

Entre mayor es la escala de una variable, mayor es su influencia en  $D$

La solución: NORMALIZAR LAS VARIABLES

¿Cómo?

Re-escalamiento

$$x'^j_i = \frac{x^j_i - \min(x^j)}{\max(x^j) - \min(x^j)}$$

$$x'^j_i \in [0,1]$$

Normalización a la media

$$x'^j_i = \frac{x^j_i - \mu^j}{\max(x^j) - \min(x^j)}$$

Estandarización (z-score)

$$x'^j_i = \frac{x^j_i - \mu^j}{\sigma^j}$$

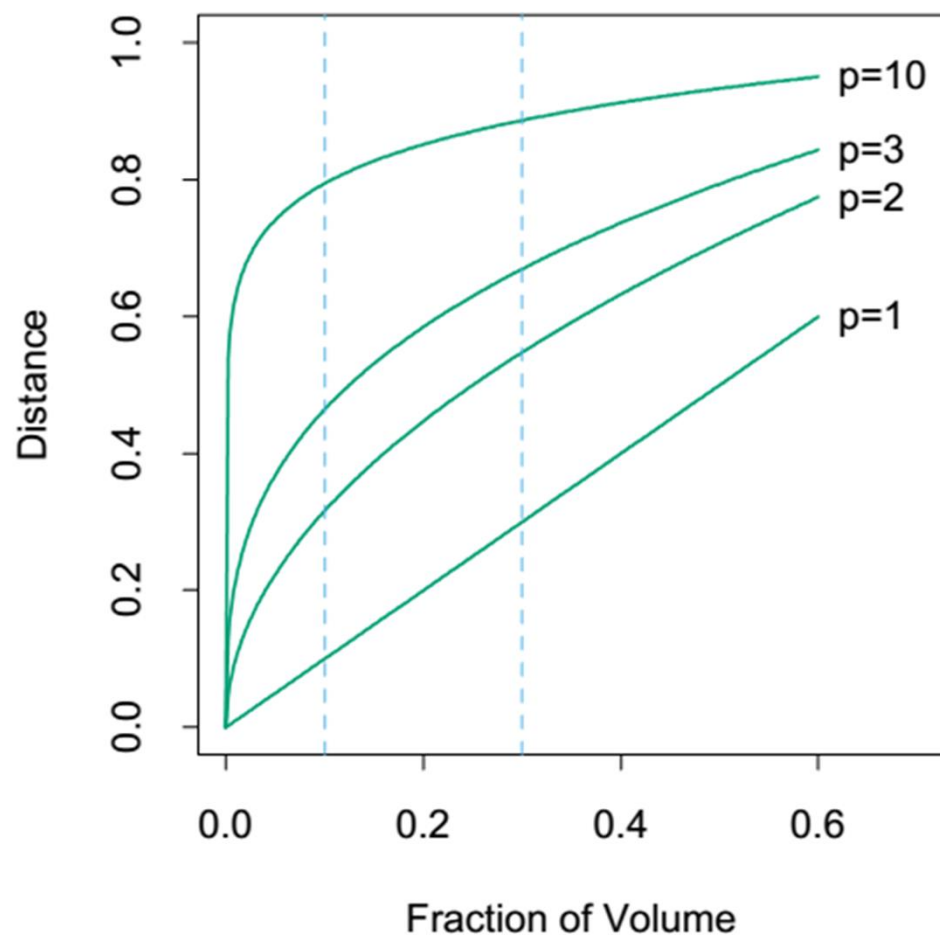
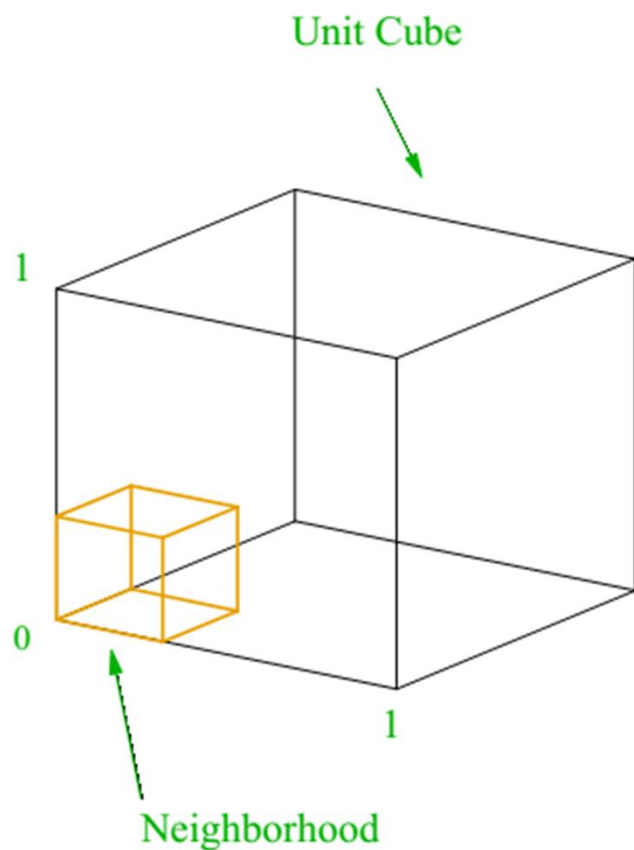
# Fortalezas y Debilidades de KNN

## Fortalezas

- Construye un modelo local por cada instancia.
- Aprender el modelo no tiene costo.
- Buen desempeño w.r.t. otras técnicas de ML.

## Debilidades

- No es aprendizaje.
- No hay un modelo global y el conocimiento es ilegible.
- No hay generalización.
- Maldición de la dimensionalidad.
- El ruido y los valores atípicos tienen un impacto negativo en el desempeño.



**FIGURE 2.6.** *The curse of dimensionality* is well illustrated by a subcubical neighborhood for uniform data in a unit cube. The figure on the right shows the side-length of the subcube needed to capture a fraction  $r$  of the volume of the data, for different dimensions  $p$ . In ten dimensions we need to cover 80% of the range of each coordinate to capture 10% of the data.