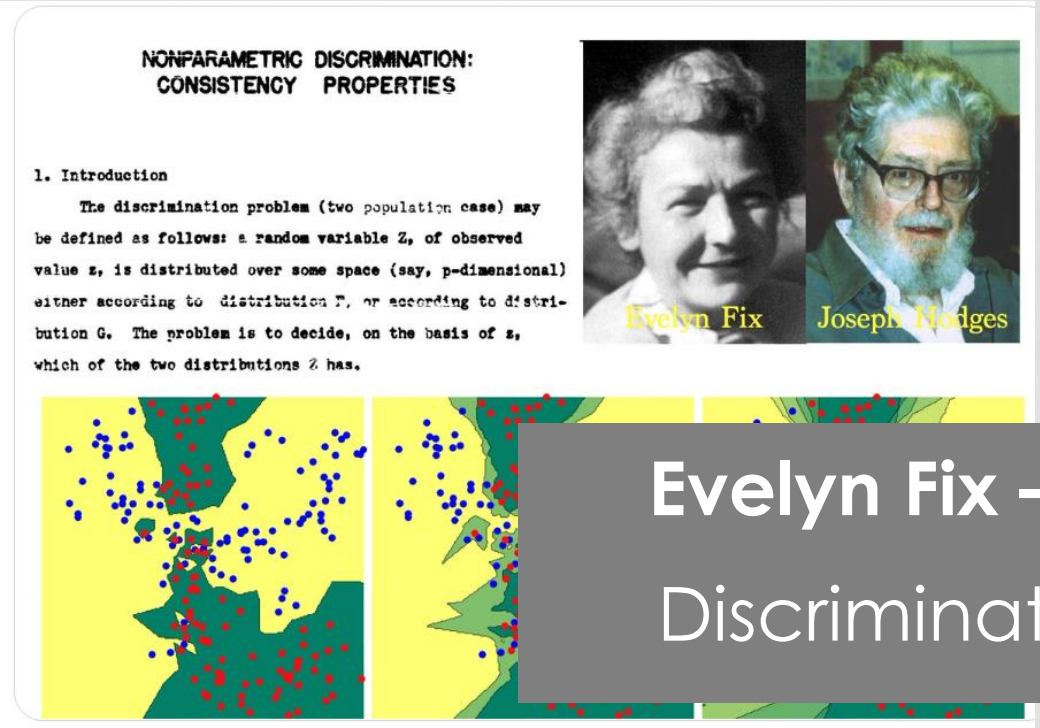
K VECINO(S) MAS CERCANO(S)

KNN = k-nearest neighbors

VECINOS MÁS CERCANOS

"Dime con quién vas y te diré quién eres"

Modelos de clasificación KNN (K-nearest-neighbors)



NONPARAMETRIC DISCRIMINATION:
CONSISTENCY PROPERTIES

1. Introduction

The discrimination problem (two population case) may be defined as follows: a random variable Z, of observed value z, is distributed over some space (say, p-dimensional) either according to distribution F, or according to distribution G. The problem is to decide, on the basis of z, which of the two distributions Z has.

Evelyn Fix – Joshep Hedges

Discriminatory analysis-1951

AGENDA

1

- **Visión general**
  - KNN para clasificación
  - Ejemplos
  - KNN para regresión

2

- **Características**
  - Similitud y distancia
  - Limitaciones
  - Ajuste de hiperparámetros: valores de K

Contenidos: Ricardo Aler Mur y Concepción García Diéguez

1. VISIÓN GENERAL

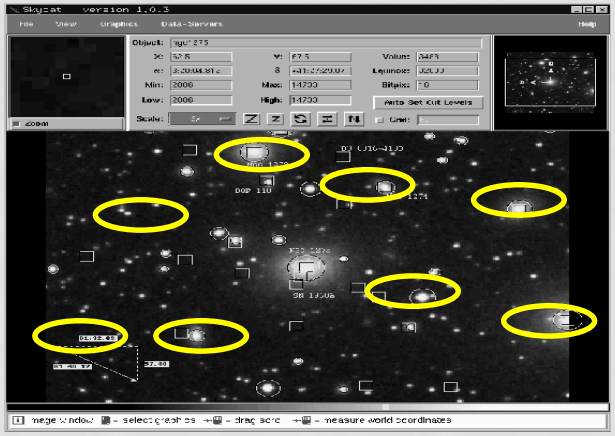
KNN PARA CLASIFICACIÓN

- Las instancias se componen de varios atributos descriptivos y un solo atributo salida (que identifica la clase a la que pertenecen)
- Clasifica cada nueva instancia en la clase que corresponda teniendo en cuenta la distancia entre datos, según tenga *k* vecinos más cerca de un grupo o de otro. Esta clase será, por tanto, la de mayor frecuencia con menores distancias.
- KNN es un algoritmo supervisado perezoso (lazy)
  - Durante el entrenamiento, sólo guarda las instancias, no construye ningún modelo (a diferencia de, por ejemplo, los árboles de decisión)
  - La clasificación se hace cuando llega la instancia de test

KNN es un método "perezoso" (lazy): el modelo son los datos

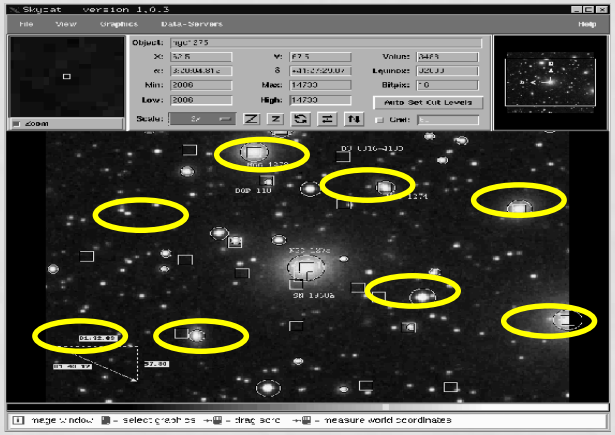
?

Datos Entrenamiento



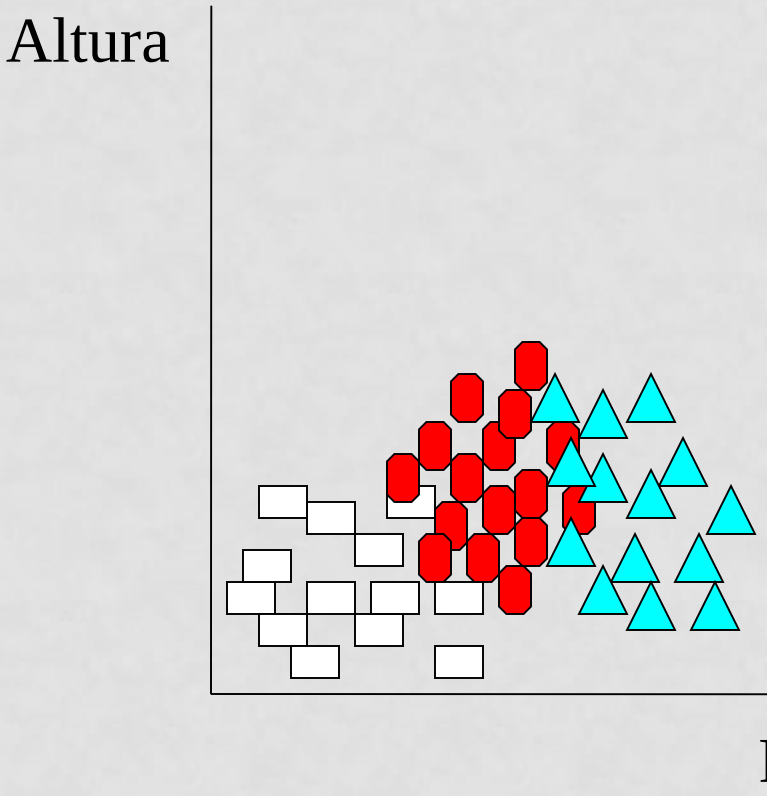
KNN

Modelo



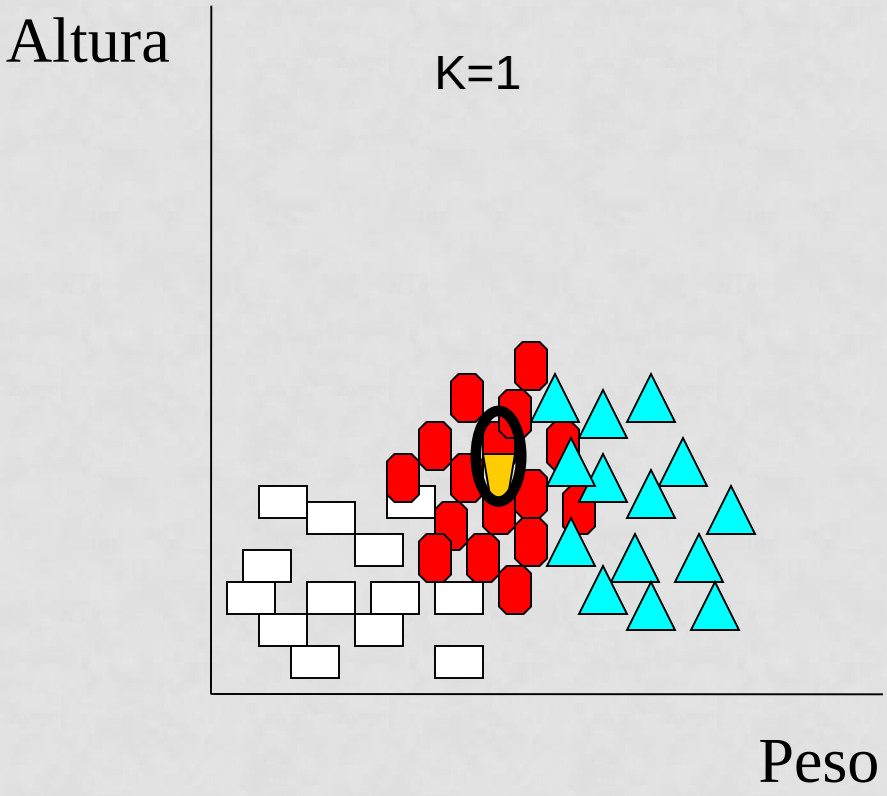
Galaxia espiral

K NEAREST NEIGHBORS (KNN)



Se guardan todos los ejemplos (lazy)

K NEAREST NEIGHBORS (KNN)



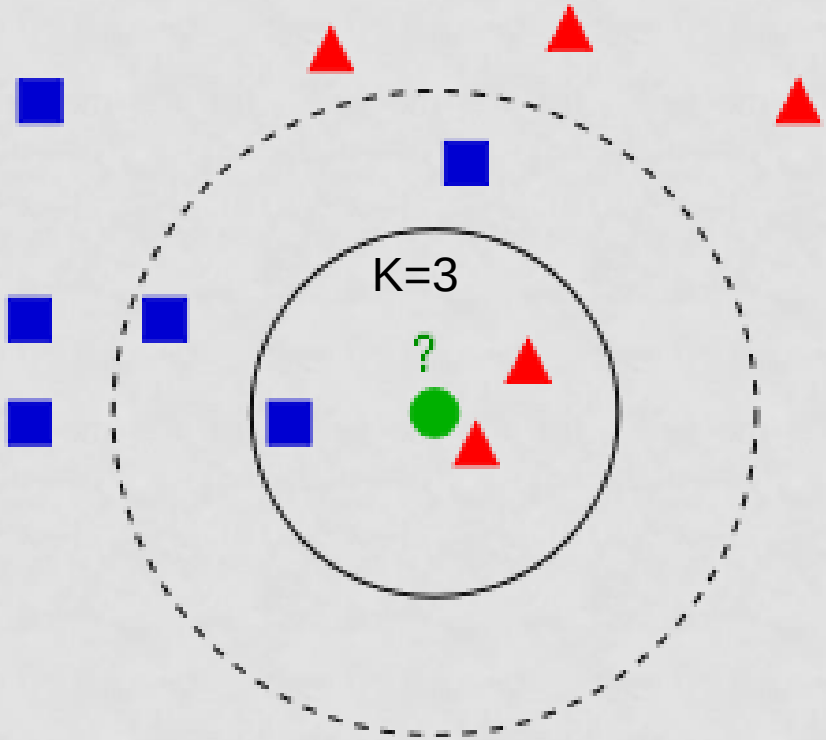
Niño

Adulto

Mayor

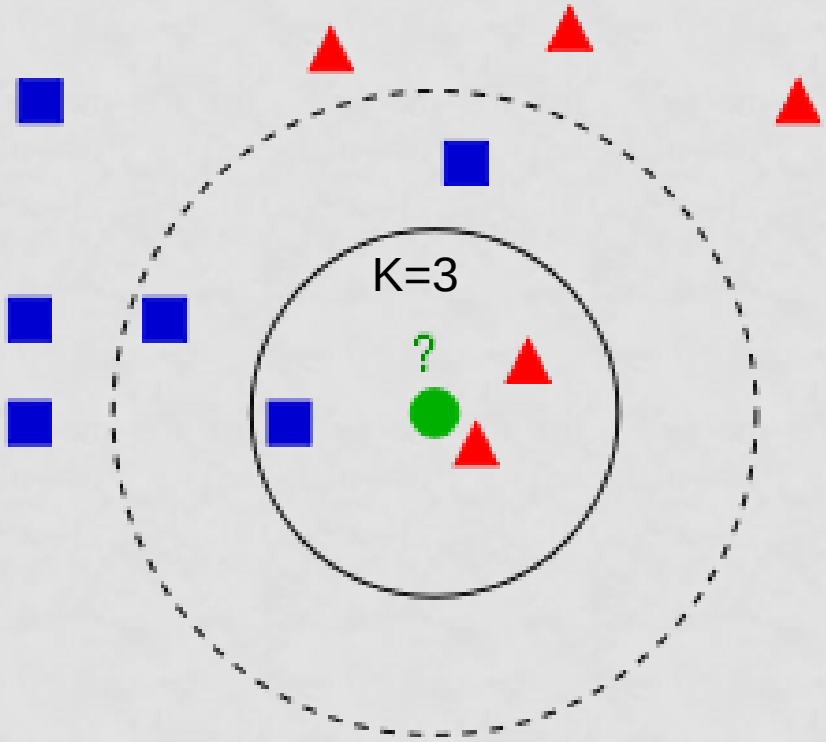
Algoritmo “lazy” (perezoso):
No se construye un modelo,
Simplemente se guardan todas las instancias

K NEAREST NEIGHBORS (KNN)



Con K=5, el modelo daría otra predicción

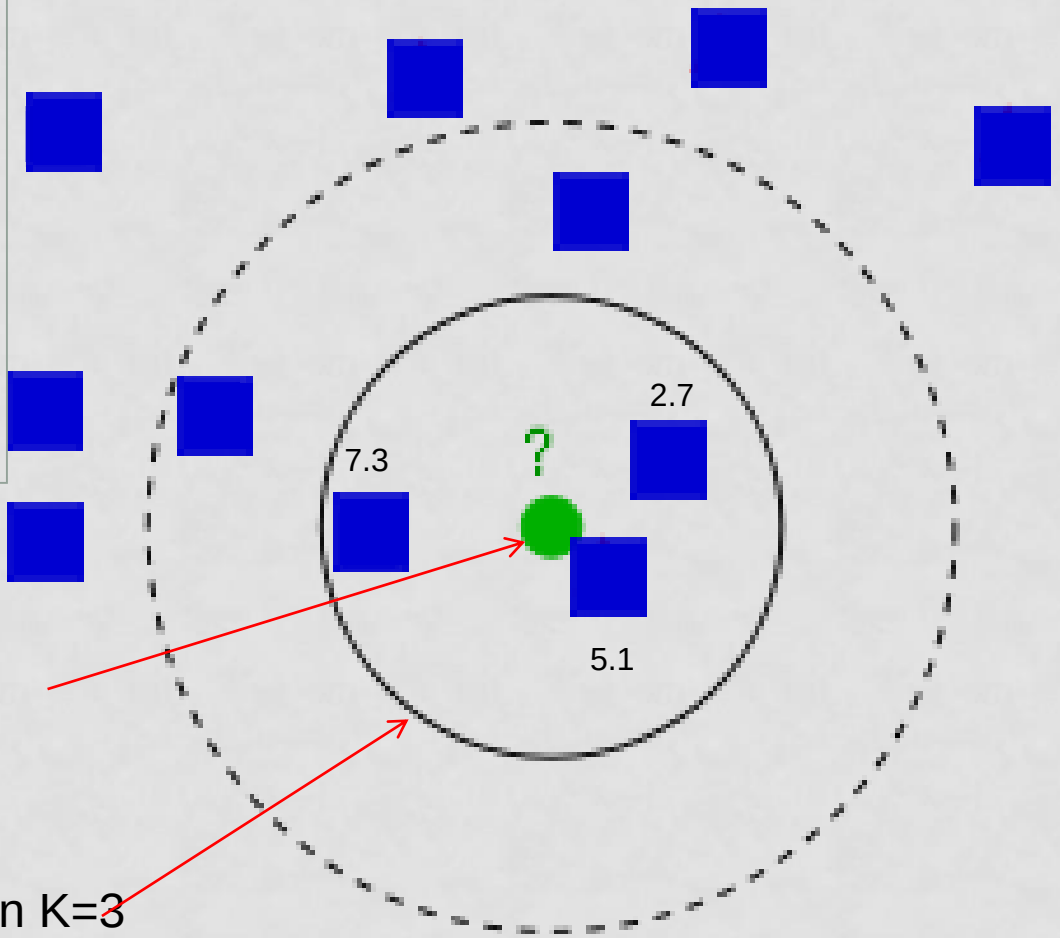
K NEAREST NEIGHBORS (KNN)



- Para evitar que los vecinos lejanos tengan mucha influencia, se puede hacer que cada vecino vote de manera inversamente proporcional a la distancia 1/d

KNN PARA REGRESIÓN

- Para predecir una variable continua de salida podemos
- calcular la media de los K vecinos más cercanos
- Para que las instancias más lejanas tengan menos importancia, se puede hacer una media ponderada por 1/d
- Se puede construir un modelo lineal con los K vecinos



Predicción = (7.3+2.7+5.1)/3

Vecindad con K=3

2. CARACTERÍSTICAS

SIMILITUD Y DISTANCIA

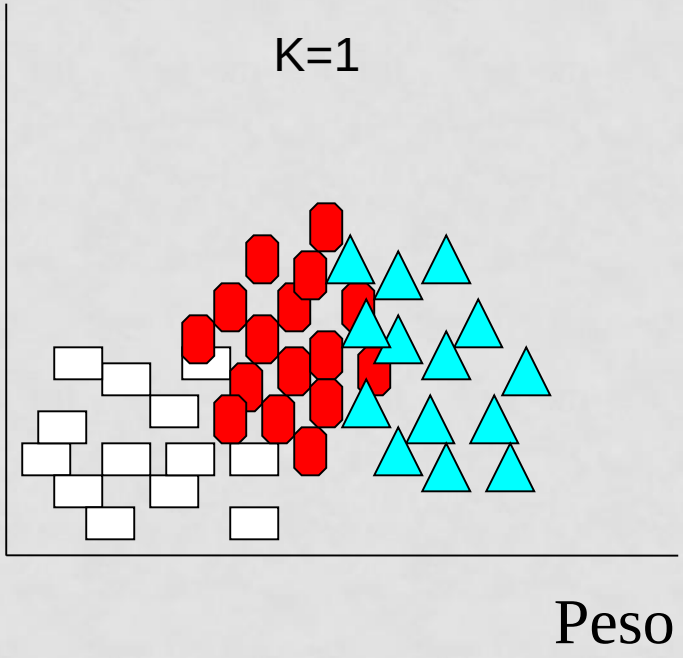
- Normalmente se usa la **distancia Euclidea**:
  - En 2D: $d(\mathbf{x_i},\mathbf{x_j})^2 = (x_{i1}-x_{j1})^2 + (x_{i2}-x_{j2})^2$ siendo $\mathbf{x_i} = (x_{i1}, x_{i2})$; $\mathbf{x_j} = (x_{j1}, x_{j2})$
  - En dD: $d(\mathbf{x_i},\mathbf{x_j})^2 = (x_{i1}-x_{j1})^2 + (x_{i2}-x_{j2})^2 + \ldots + (x_{id}-x_{jd})^2$
  - Es necesario re-escalar (**normalizar**) los atributos para que atributos con mucho rango no tengan mas peso que los demás (preproceso):
    - Rango (*minmax*): $x'_{1j} = (x_{1j}-min_j)/(max_j-min_i)$
  - Es aconsejable centrar y reducir (**estandarizar**) los atributos para que no entorpezcan los resultados del algoritmo (preproceso)
    - Estandarización: $x'_{1j} = (x_{1j}-\mu_j)/\sigma_j$
- Si los atributos son nominales, usar **distancia de Hamming**:
  - Si el atributo e es nominal (o discreto o categórico), en lugar del componente $(x_{ie}-x_{je})^2$ se usa:
    - $\delta(x_{ie}, x_{je})=0$ si $x_{ie}=x_{je}$; $\delta(x_{ie}, x_{je})=1$ en caso contrario
  - También, variables "**dummy**" o "**one-hot**" (preproceso)

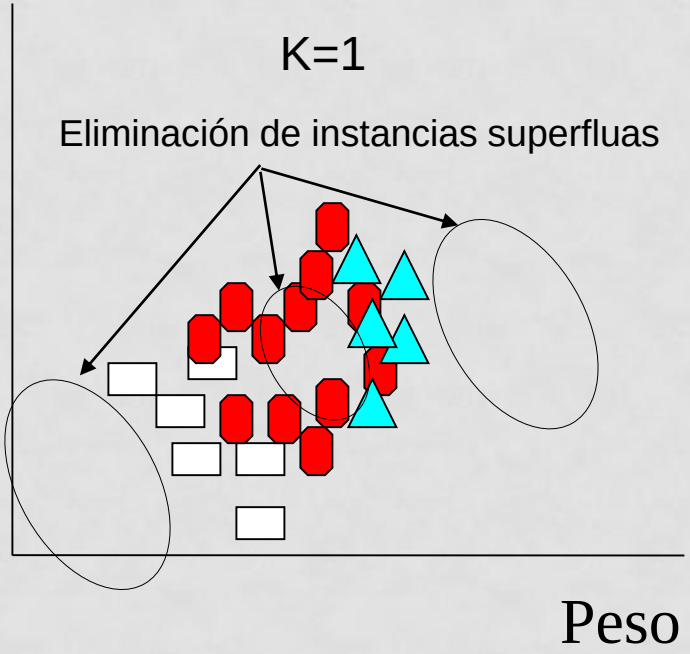
LIMITACIONES DE KNN

- Lento, si hay muchos datos de entrenamiento (en almacenamiento y en tiempo):
  -  Eliminación de instancias superfluas (fase de preproceso)
- Muy sensible a los atributos irrelevantes:
  -  Selección de atributos (fase de preproceso)
- Muy sensible al ruido:
  -  Ajuste del hiper-parámetro del número de vecinos (K)

LIMITACIONES KNN: ELIMINAR INSTANCIAS SUPERFLUAS

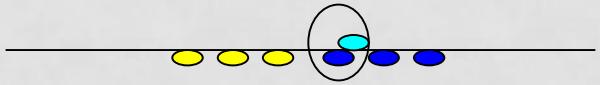
- Hay instancias superfluas: no son necesarias para clasificar. Si las borramos se decrementará el tiempo de clasificación





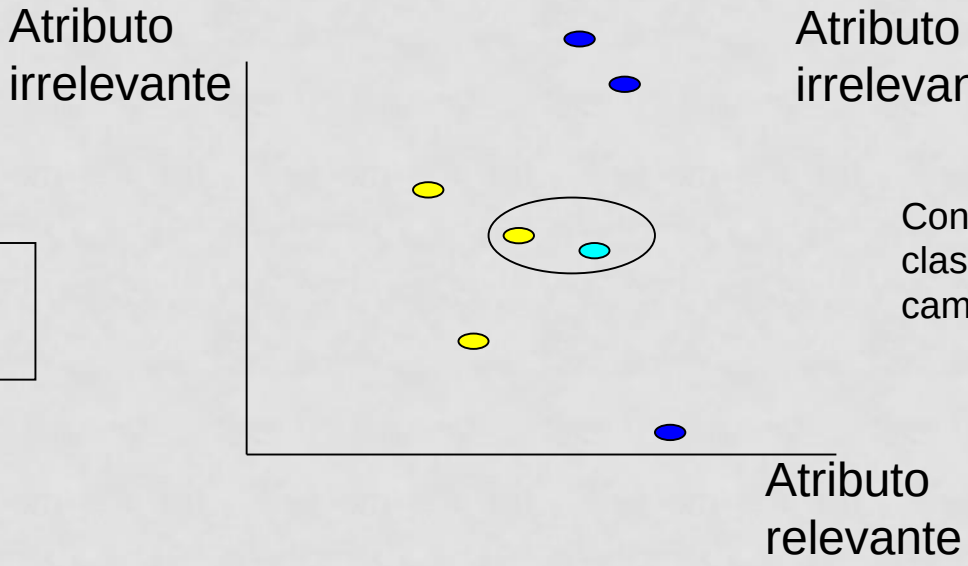
LIMITACIONES KNN: ATRIBUTOS IRRELEVANTES

0 atributos irrelevantes



Con el atributo relevante, se clasifica bien

1 atributo irrelevante



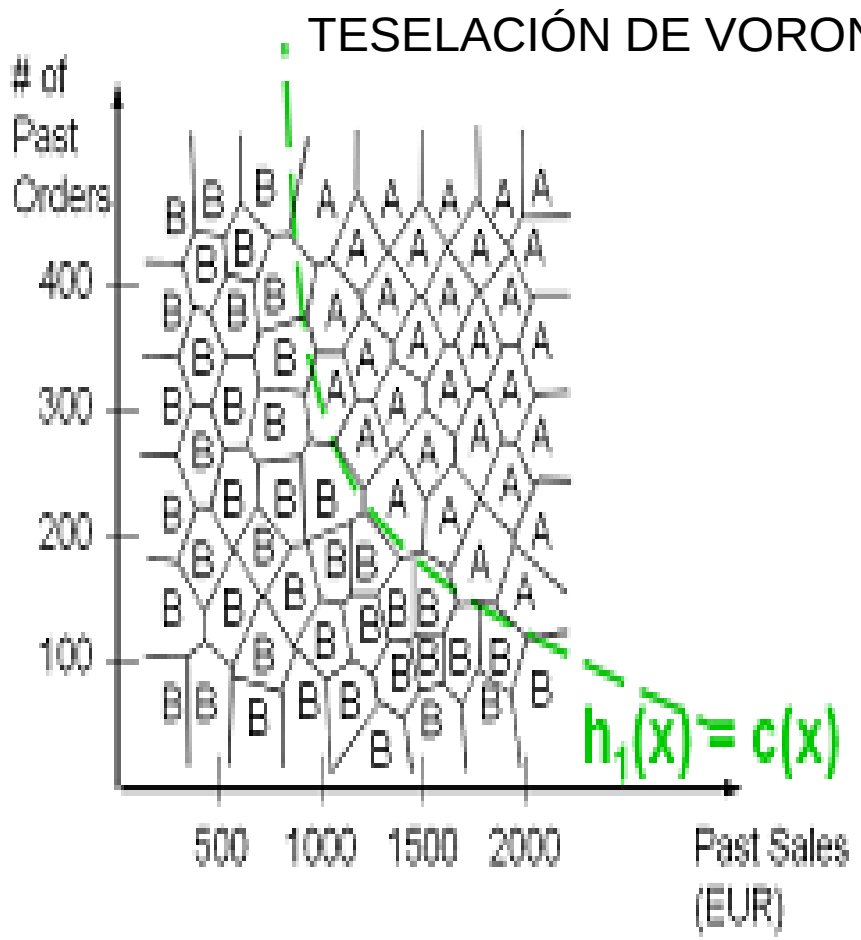
Con el atributo irrelevante, se clasifica mal (las distancias cambian)

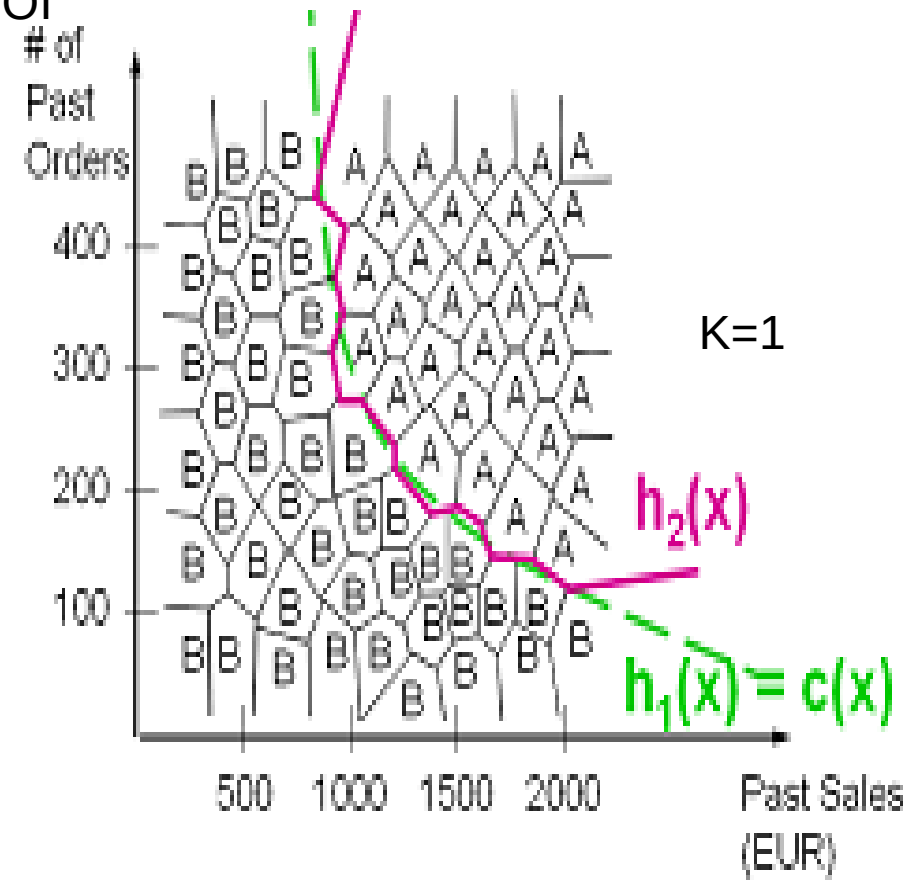
Vecino mas cercano k=1

AJUSTE DE HIPERPARÁMETROS: VALORES DE K

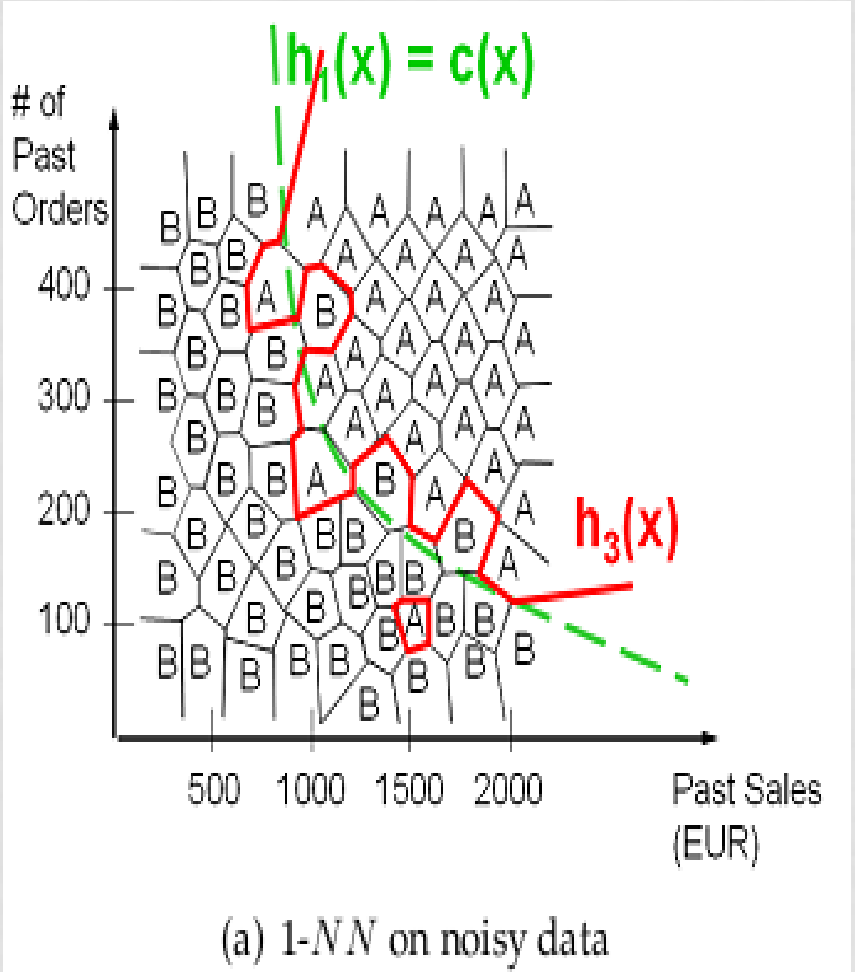
- K es el hiper-parámetro principal de KNN. La mejor elección de K depende fundamentalmente de los datos de partida. Un buen K puede ser seleccionado mediante optimización de uso
- Con K=1, las instancias con características irrelevantes, con ruido o solape entre clases tienen mucha influencia
- Con K>1, se consideran mas vecinos y el ruido pierde influencia (es como hacer una promediado)
  - Si k es muy alto, se pierde la idea de localidad
    - ¿En que se convierte KNN si K == número de datos?
  - Si hay dos clases,
    - ¿Qué riesgo hay de usar un K par?

INFLUENCIA DE DATOS CON RUIDO

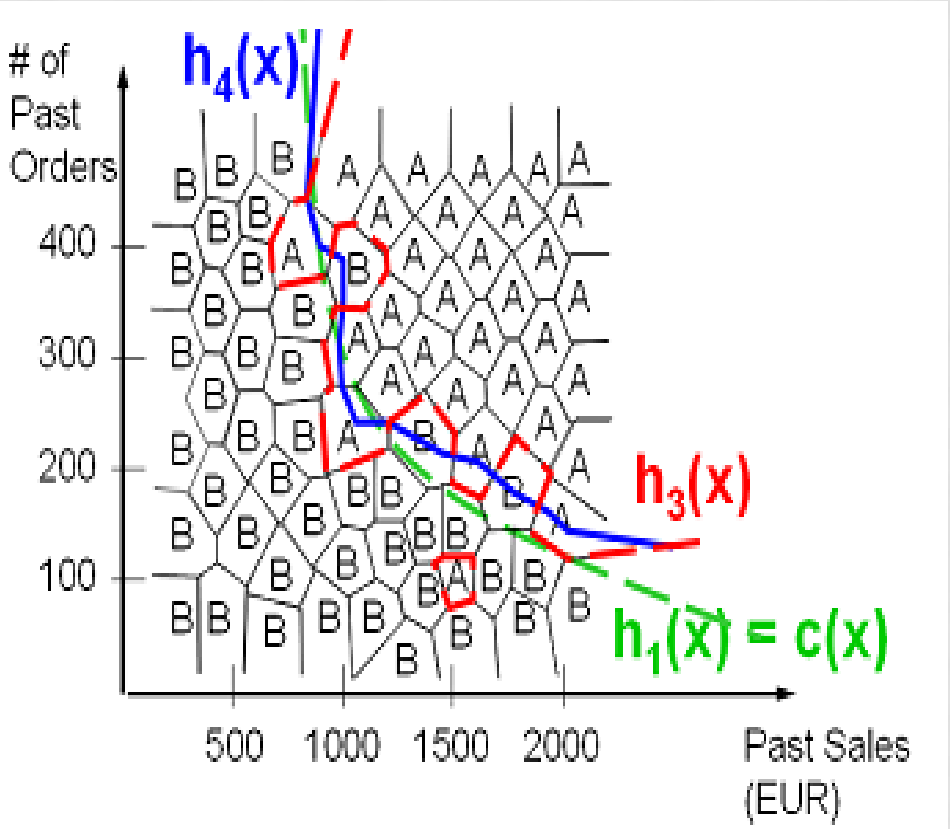




INFLUENCIA DE DATOS CON RUIDO



(a) 1-NN on noisy data



(b) 3-NN and noisy data