VERDAD, BELLEZA, PROBIDAD

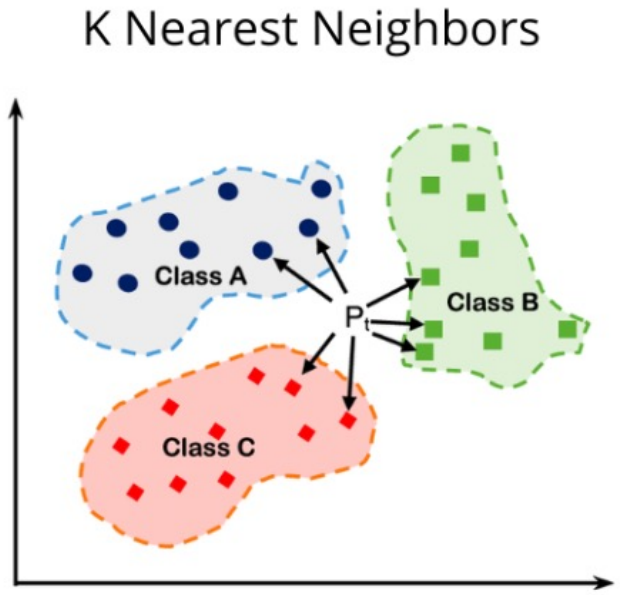
Facultad de Ingeniería y Ciencias

K-vecinos más cercanos

Dr. José Lázaro Martínez Rodríguez

Introducción

- El algoritmo de k vecinos más cercanos, también conocido como KNN o k-NN, es un clasificador de aprendizaje supervisado no paramétrico, que utiliza la proximidad para hacer clasificaciones o predicciones sobre la agrupación de un punto de datos individual.



K-Vecinos más Cercanos

- Idea: ***“Dime con quién vas y te diré quién eres”***
- Es un algoritmo de aprendizaje supervisado
- Se usa en tareas de clasificación y regresión
- El algoritmo clasifica cada dato nuevo en el grupo que corresponda, según tenga ***k*** vecinos más cerca de un grupo o de otro.

K-Vecinos más Cercanos

- No produce un modelo a partir de los datos de entrenamiento
- El aprendizaje sucede en el momento en que se prueban los datos de test
- Se selecciona un valor de k
- El algoritmo selecciona las k instancias más cercanas y se asigna la clase más frecuente de entre las k instancias seleccionadas.
- El método es no paramétrico

K-Vecinos más Cercanos

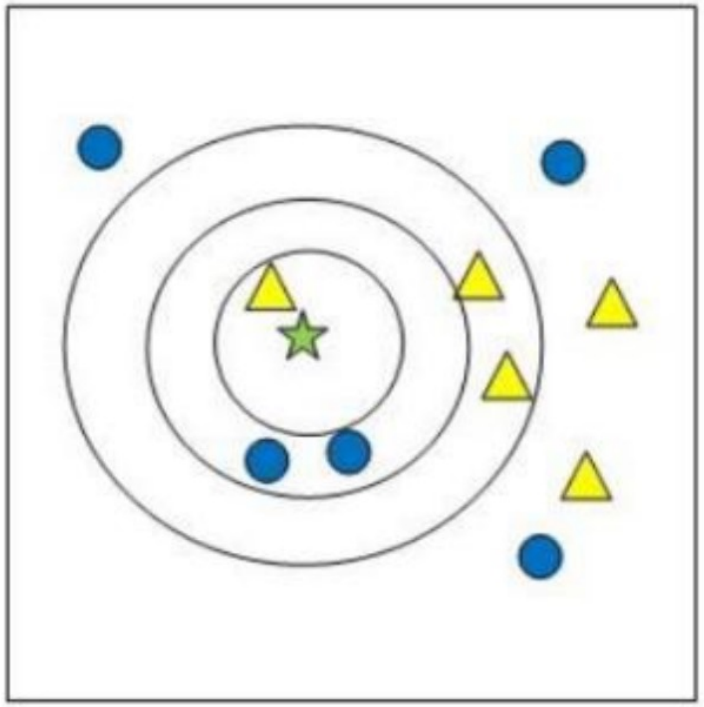
COMIENZO
  Entrada: $D = \{(\mathbf{x}_1, c_1), \ldots, (\mathbf{x}_N, c_N)\}$
    $\mathbf{x} = (x_1, \ldots, x_n)$ nuevo caso a clasificar
  PARA todo objeto ya clasificado (x_i, c_i)
    calcular $d_i = d(\mathbf{x}_i, \mathbf{x})$
  Ordenar $d_i (i = 1, \ldots, N)$ en orden ascendente
  Quedarnos con los K casos $D_{\mathbf{x}}^K$ ya clasificados más cercanos a $\mathbf{x}$
  Asignar a $\mathbf{x}$ la clase más frecuente en $D_{\mathbf{x}}^K$
FIN

Algorithm (kNN):

1. Find k examples $\{\mathbf{x}^{(i)}, t^{(i)}\}$ closest to the test instance $\mathbf{x}$
2. Classification output is majority class

$$y = arg \max_{t^{(z)}} \sum_{r=1}^{k} \delta(t^{(z)}, t^{(r)})$$

K-Vecinos más Cercanos

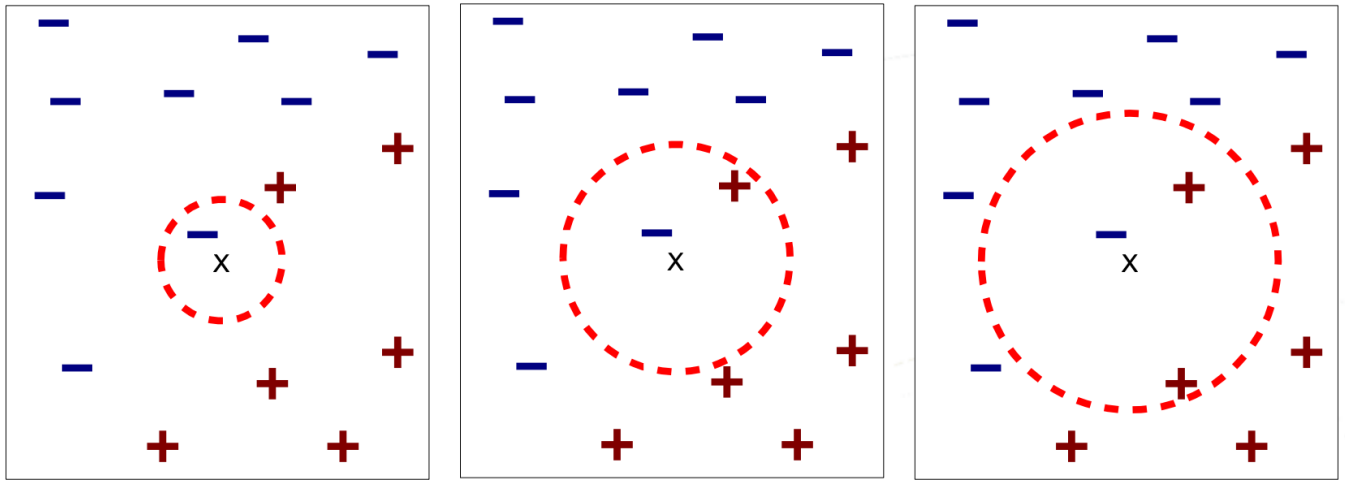


¿Qué clase será asignada a la instancia si k = 1?

¿Qué clase será asignada a la instancia si k = 3?

K-Vecinos más Cercanos
¿Cómo seleccionar k?

- Si k es muy pequeño el modelo será muy sensitivo a puntos que son atípicos o que son ruido (datos corruptos)
- Si k es muy grande, el modelo tiende a asignar siempre la clase más grande



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

¿Cómo trabaja el algoritmo Knn?

- El algoritmo K-NN compara una nueva entrada de datos con los valores de un conjunto de datos determinado (con diferentes clases o categorías).

- En función de su cercanía o similitud en un rango determinado (K) de vecinos, el algoritmo asigna los nuevos datos a una clase o categoría del conjunto de datos (datos de entrenamiento).

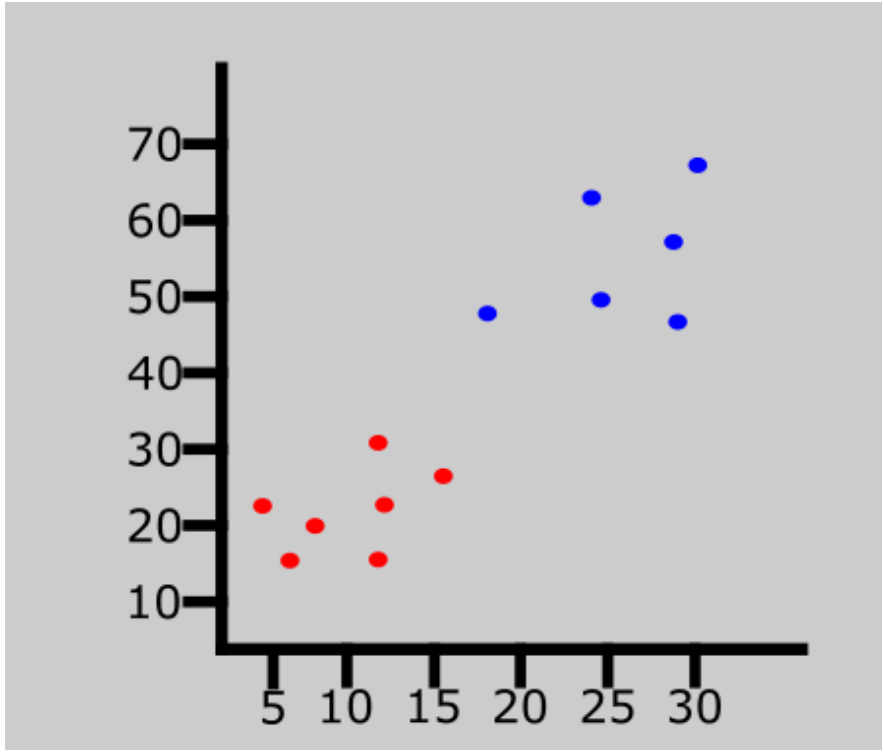
¿Cómo trabaja el algoritmo Knn?

- Paso #1 - Asigne un valor a K.
- Paso #2 - Calcule la distancia entre la nueva entrada de datos y todas las demás entradas de datos existentes. Ordénelas en orden ascendente.
- Paso #3 - Encuentre los K vecinos más cercanos a la nueva entrada basándose en las distancias calculadas.
- Paso #4 - Asigne la nueva entrada de datos a la clase mayoritaria en los vecinos más cercanos.

Ejemplo con diagramas

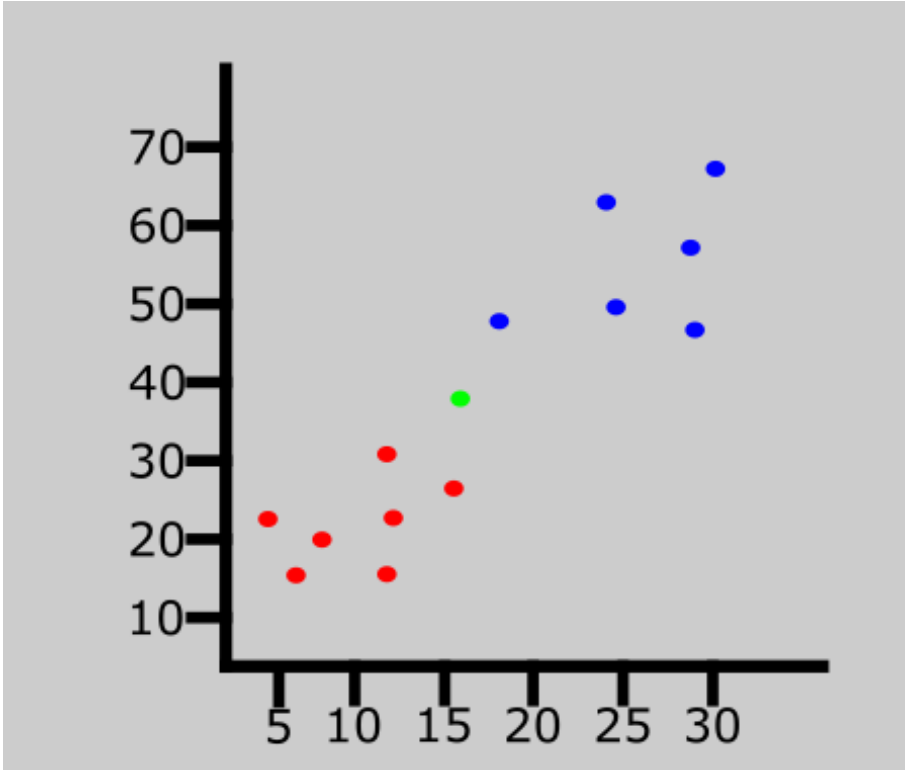
- Considere el siguiente diagrama

- Representa un conjunto de datos que consiste en dos clases (rojo y azul)



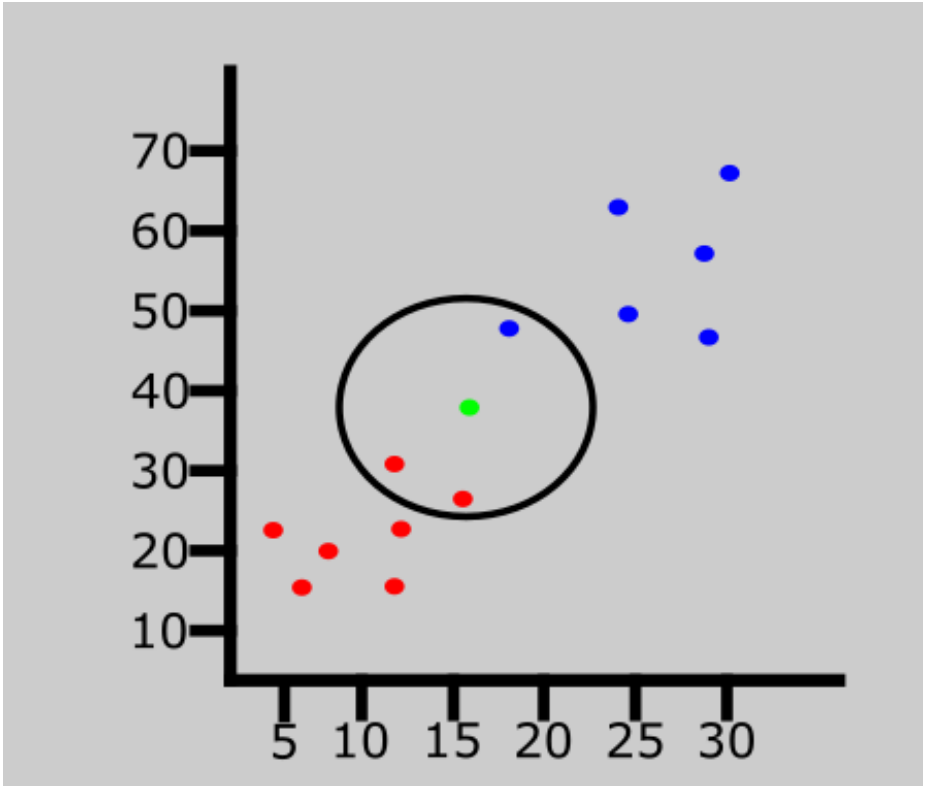
Ejemplo con diagramas

- Se ha introducido un nuevo elemento al conjunto de datos.
- El cual se representa por el punto verde



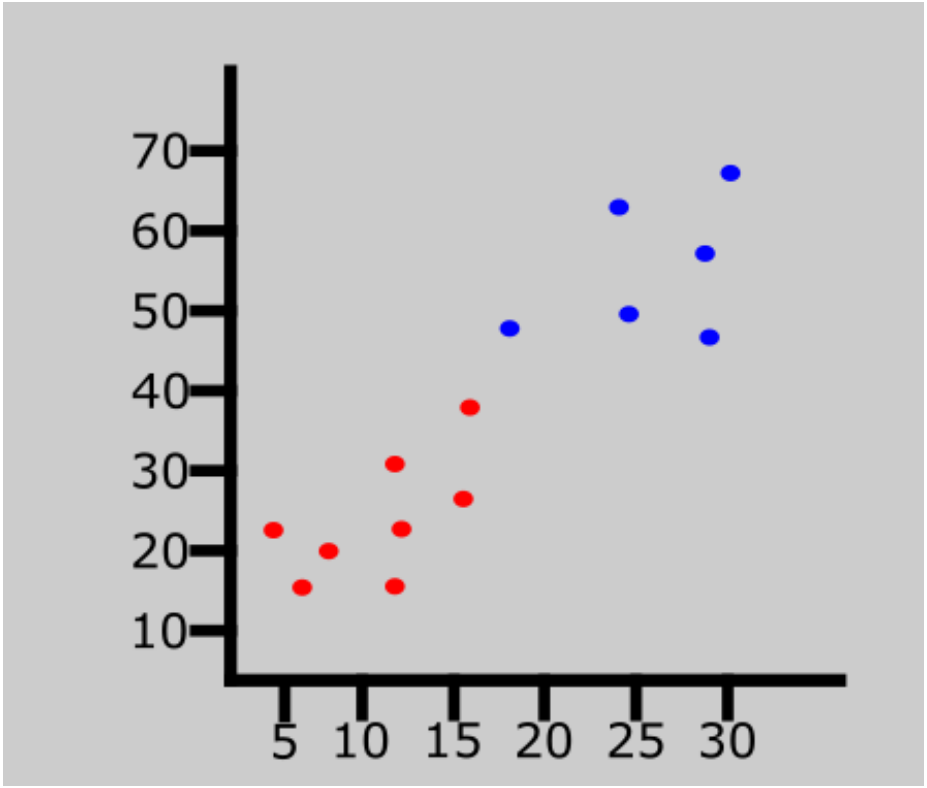
Ejemplo con diagramas

- A continuación, asignaremos un valor a K que denota el número de vecinos a tener en cuenta antes de clasificar la nueva entrada de datos.
- Supongamos que el valor de K es 3.
- Como el valor de K es 3, el algoritmo sólo tendrá en cuenta los 3 vecinos más próximos al punto verde (nueva entrada).



Ejemplo con diagramas

- De los 3 vecinos más cercanos del diagrama anterior, la clase mayoritaria es la roja, por lo que la nueva entrada se asignará a esa clase.



Ejemplo con dataset

- En este ejemplo asignaremos datos
- La tabla representa nuestro conjunto de datos. Tenemos dos columnas: Brillo y Saturación.
- Cada fila de la tabla tiene una clase de Rojo o Azul.

BRIGHTNESS	SATURATION	CLASS
40	20	Red
50	50	Blue
60	90	Blue
10	25	Red
70	70	Blue
60	10	Red
25	80	Blue

¿Cómo calculamos la distancia?

- Recordar que hemos estudiado dos formas
- $d(u,v) = \|u - v\| = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \cdots + (a_n - b_n)^2}$
- $\cos\theta = \frac{u \cdot v}{\|u\|\|v\|}$

- Entonces con estas podemos sacar la distancia o cercanía de los puntos existentes con respecto al nuevo punto

¿Cómo calculamos la distancia?

- Por ejemplo, si tenemos un nuevo dato

BRIGHTNESS	SATURATION	CLASS
20	35	?

- Tenemos una nueva entrada, pero aún no tiene clase. Para conocer su clase, tenemos que calcular la distancia de la nueva entrada a otras entradas en el conjunto de datos utilizando una de las medidas vistas
- $\sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2} = \sqrt{(20 - b_1)^2 + (35 - b_2)^2}$

b se sustituye de la tabla de datos inicial, uno a uno

¿Cómo calculamos la distancia?

- Hagamos los cálculos con los primeros tres puntos
- Para el primer registro obtenemos distancia "$d1$"

BRIGHTNESS	SATURATION	CLASS
40	20	Red

- $d1 = \sqrt{(20-40)^2 + (35-20)^2}$
- $= \sqrt{400 + 225}$
- $= \sqrt{625}$
- $= \mathbf{25}$

¿Cómo calculamos la distancia?

- Ahora conocemos la distancia entre la nueva entrada de datos y la primera entrada de la tabla. Actualicemos la tabla.

BRIGHTNESS	SATURATION	CLASS	DISTANCE
40	20	Red	25
50	50	Blue	?
60	90	Blue	?
10	25	Red	?
70	70	Blue	?
60	10	Red	?
25	80	Blue	?

¿Cómo calculamos la distancia?

- Para el segundo registro obtenemos distancia "$d2$"

BRIGHTNESS	SATURATION	CLASS	DISTANCE
50	50	Blue	?

- $d2 = \sqrt{(20-50)^2 + (35-50)^2}$
- $= \sqrt{900 + 225}$
- $= \sqrt{1125}$
- $= \mathbf{33.54}$

¿Cómo calculamos la distancia?

- Actualicemos la tabla.

BRIGHTNESS	SATURATION	CLASS	DISTANCE
40	20	Red	25
50	50	Blue	33.54
60	90	Blue	?
10	25	Red	?
70	70	Blue	?
60	10	Red	?
25	80	Blue	?

¿Cómo calculamos la distancia?

- Para el tercer registro obtenemos distancia "$d3$"

BRIGHTNESS	SATURATION	CLASS	DISTANCE
60	90	Blue	?

- $d3 = \sqrt{(20-60)^2 + (35-90)^2}$
- $= \sqrt{1600 + 3025}$
- $= \sqrt{4625}$
- $= \mathbf{68.01}$

¿Cómo calculamos la distancia?

- Actualicemos la tabla.
- Llegados a este punto, deberías entender cómo funciona el cálculo. Intenta calcular la distancia de las cuatro últimas filas.

BRIGHTNESS	SATURATION	CLASS	DISTANCE
40	20	Red	25
50	50	Blue	33.54
60	90	Blue	68.01
10	25	Red	?
70	70	Blue	?
60	10	Red	?
25	80	Blue	?

¿Cómo calculamos la distancia?

- Este es el aspecto que tendrá la tabla una vez calculadas todas las distancias:

BRIGHTNESS	SATURATION	CLASS	DISTANCE
40	20	Red	25
50	50	Blue	33.54
60	90	Blue	68.01
10	25	Red	10
70	70	Blue	61.03
60	10	Red	47.17
25	80	Blue	45

KNN

- Reordenemos las distancias en orden ascendente:

BRIGHTNESS	SATURATION	CLASS	DISTANCE
10	25	Red	10
40	20	Red	25
50	50	Blue	33.54
25	80	Blue	45
60	10	Red	47.17
70	70	Blue	61.03
60	90	Blue	68.01

KNN

- Si elegimos 5 como valor de K, sólo consideraremos las cinco primeras filas. Es decir:

BRIGHTNESS	SATURATION	CLASS	DISTANCE
10	25	Red	10
40	20	Red	25
50	50	Blue	33.54
25	80	Blue	45
60	10	Red	47.17

KNN

- Como puede ver arriba, la clase mayoritaria entre los 5 vecinos más cercanos a la nueva entrada es Roja. Por lo tanto, clasificaremos la nueva entrada como Roja.
- Esta es la tabla actualizada
- El valor de K debe ser impar, por ejemplo 3,5,7,9
- ¿Qué clase corresponde para K= 3 y para K=7 en el ejemplo anterior?

BRIGHTNESS	SATURATION	CLASS
40	20	Red
50	50	Blue
60	90	Blue
10	25	Red
70	70	Blue
60	10	Red
25	80	Blue
20	35	Red

Ejercicio

- La tabla adjunta contiene 8 casos bidimensionales que constituyen el conjunto de entrenamiento para un clasificador k-NN.

Caso	X1	X2	Clase
1	2	0	0
2	4	4	1
3	1	1	0
4	2	4	1
5	2	2	0
6	2	3	1
7	3	4	0
8	3	3	1

- Clasifica el caso (2.5, 2.5) a partir del conjunto de entrenamiento considerando k = 1, k=3, k=5.

Ejercicio

- La tabla adjunta contiene 8 casos bidimensionales que constituyen el conjunto de entrenamiento para un clasificador k-NN.

Caso	X1	X2	Clase
1	2	0	0
2	4	4	1
3	1	1	0
4	2	4	1
5	2	2	0
6	2	3	1
7	3	4	0
8	3	3	1

- Clasifica el caso (2.5, 2.5) a partir del conjunto de entrenamiento considerando k = 1, k=3, k=5.
- Utilice dos métricas, distancia euclidiana y medida de cosenos.
  - Es decir, obtendrá dos distancias por cada punto
  - Verifique qué clase se asigna utilizando las distintas distancias

Ejercicio

- La tabla adjunta contiene 6 casos bidimensionales que constituyen el conjunto de entrenamiento para un clasificador k-NN. Los dos últimos casos -numerados como 7 y 8- forman el conjunto de prueba.

Caso	X	Y	Clase
1	1	1	0
2	2	4	0
3	3	2	0
4	3	5	0
5	4	4	0
6	4	7	1
7	6	4	?
8	6	6	?

- Construye el clasificador k-NN a partir de la distancia euclidiana y de dos valores distintos de k (k= 3; 5).