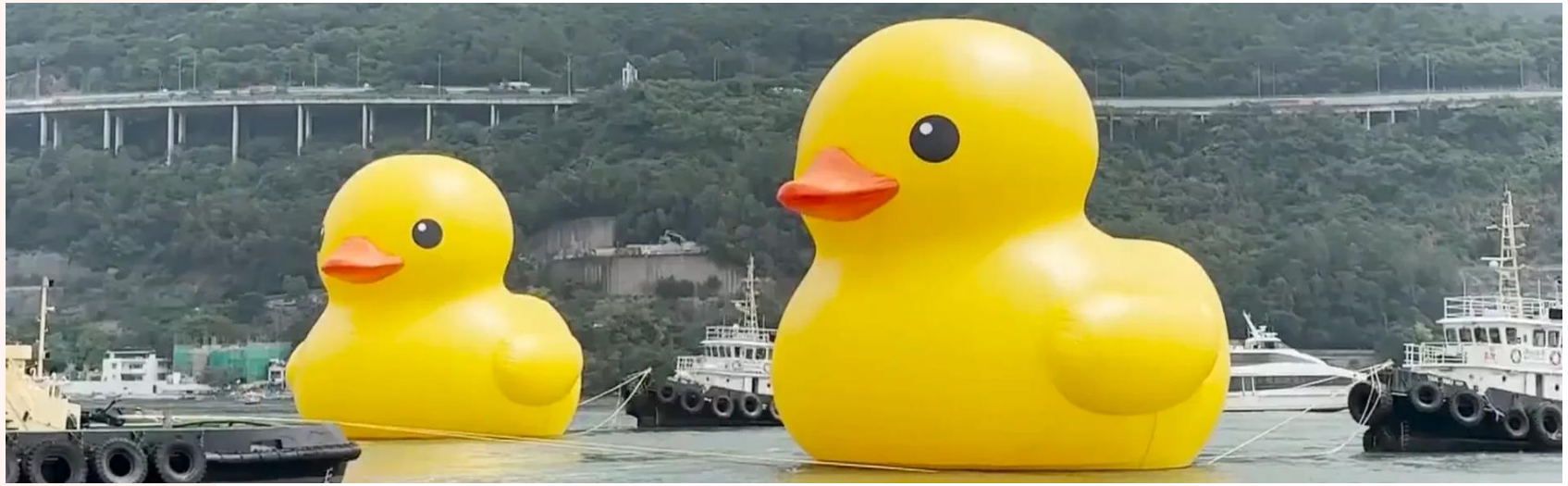
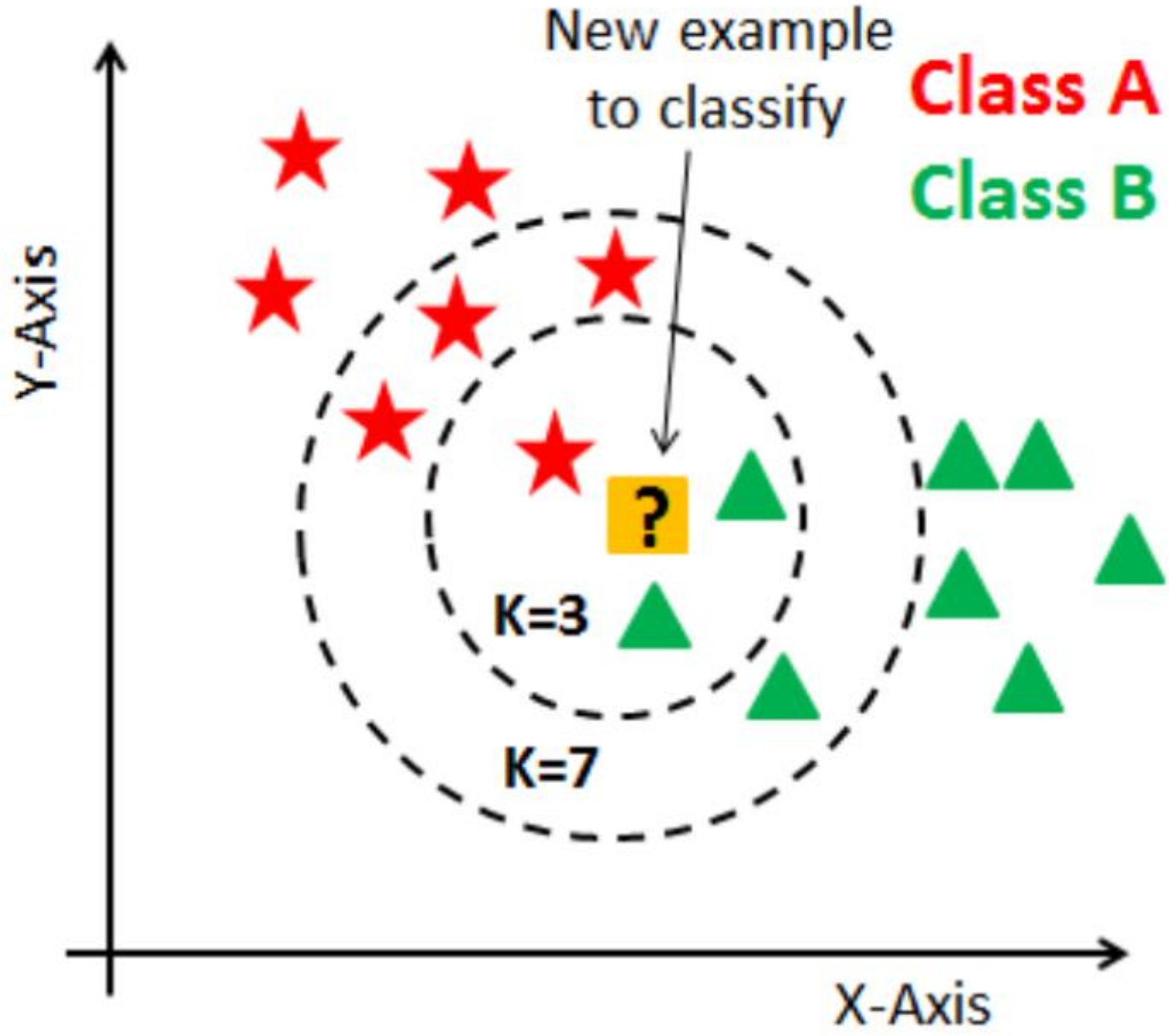
K-Nearest Neighbors

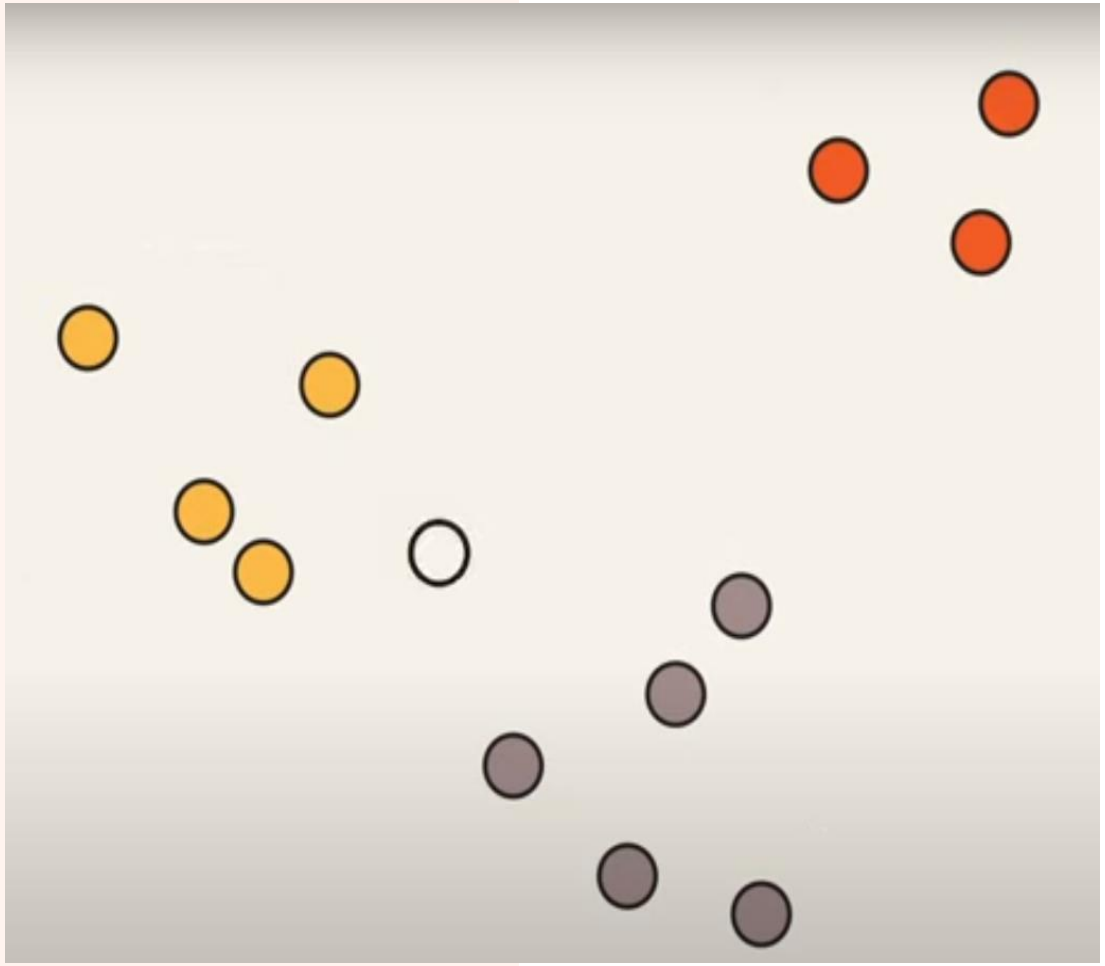
Mirjam Nilsson

Idea básica



Si camina como un pato, grazna como un pato, entonces probablemente sea un pato.


New example
to classify
Class A
Class B
Y-Axis
?
K=3
K=7
X-Axis

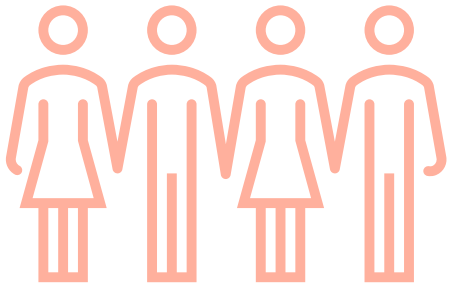


Definición

K-Vecinos más cercanos (KNN) es un algoritmo simple y no paramétrico que se utiliza en clasificación y regresión.

Clasifica un punto de datos en función de cómo se clasifican sus vecinos. La observación se asigna a la clase más común entre sus K vecinos más cercanos.

¿Qué se necesita? - Hiperpárametros



K

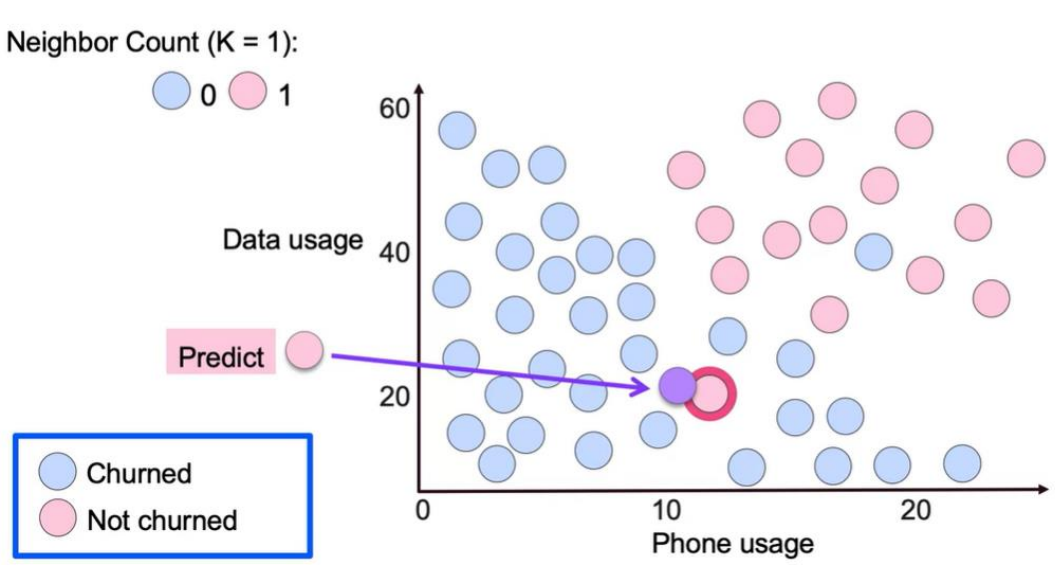
Número de vecinos

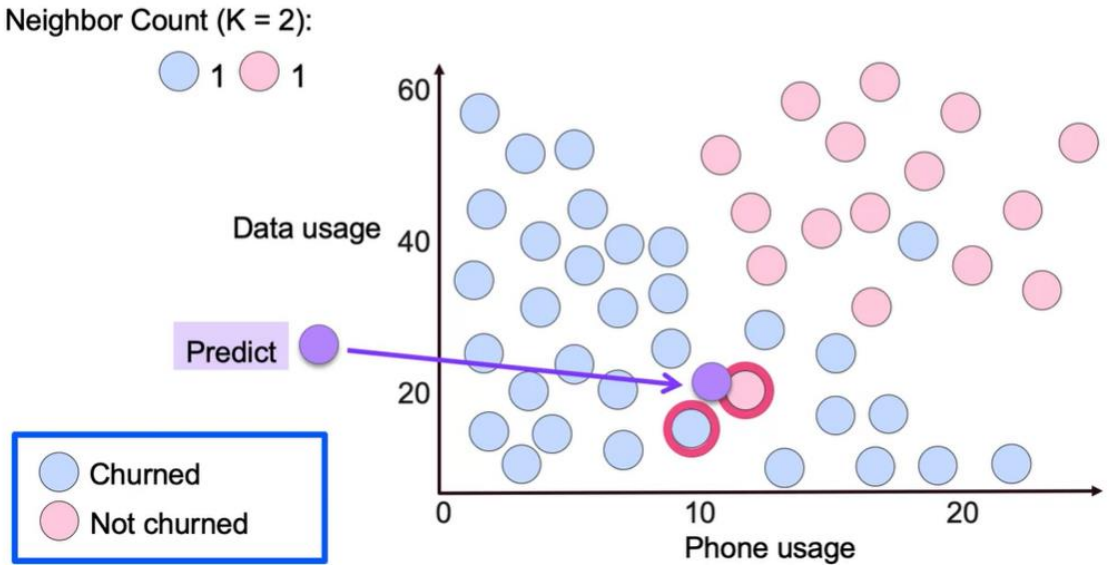
Cómo medir la distancia

Pesos

Hay vecinos que deban ser considerados más que otros, tal vez por su nivel de cercanía

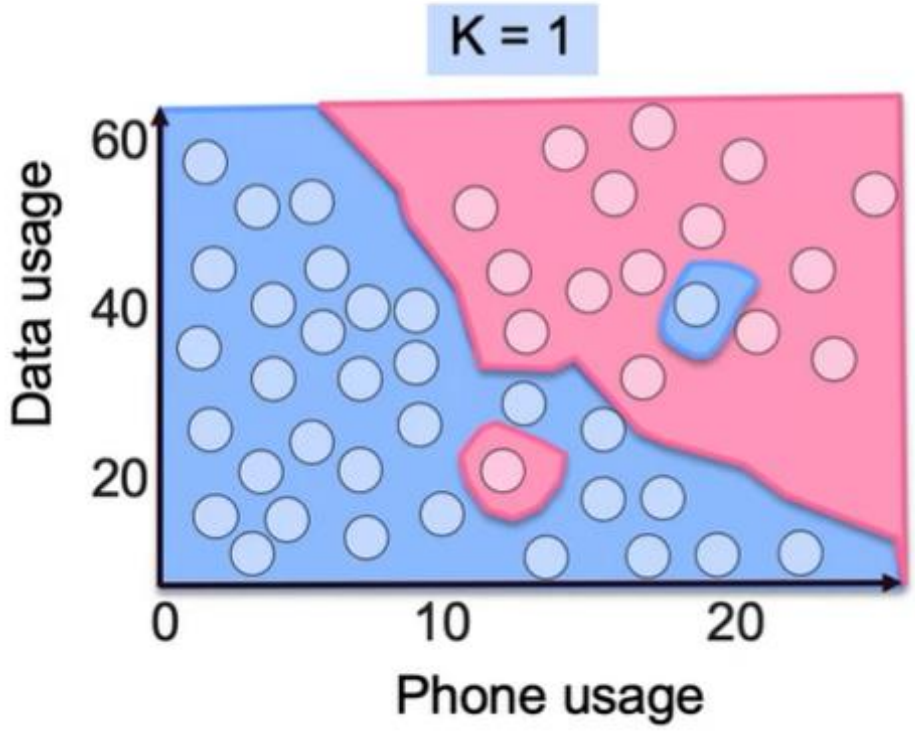
1. Eligiendo el K correcto



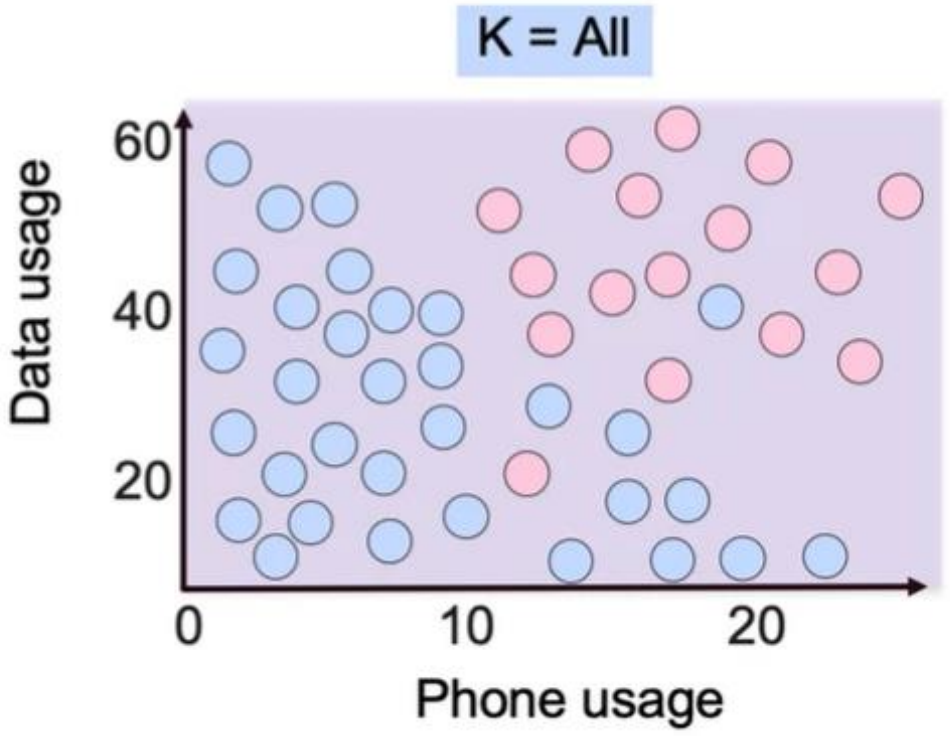


1. Eligiendo el K correcto

K pequeño - Overfitting



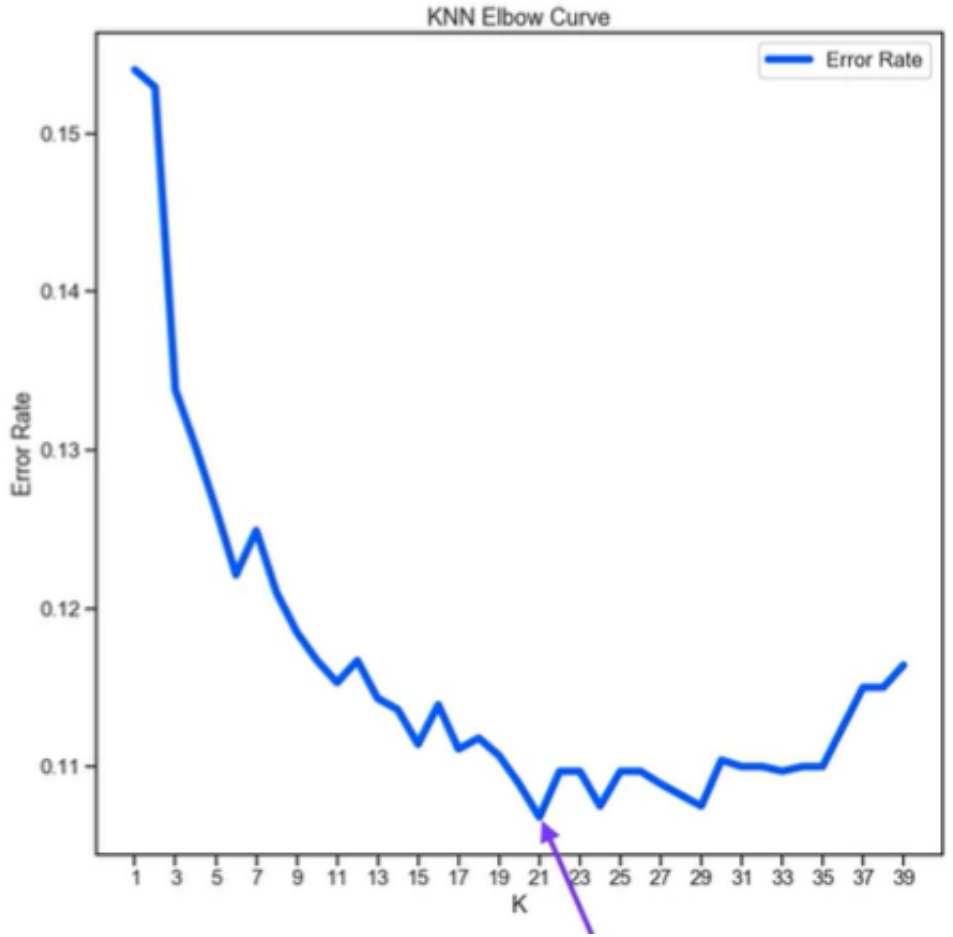
K grande - Underfitting



1. Eligiendo el K correcto

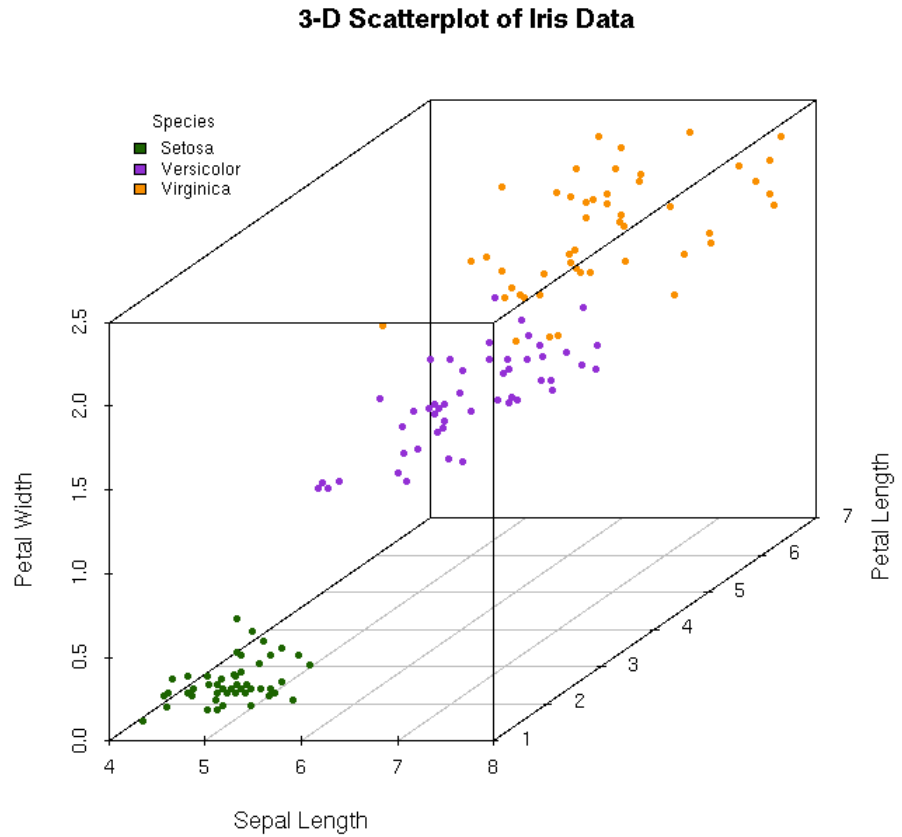
1. Puedes utilizar validación cruzada pero a la larga todo dependerá de la **métrica de error** de tu interés.
2. Método del “codo”

1. Eligiendo el K correcto



El 'método del codo' implica **graficar la tasa de error frente a diferentes valores de K** y elegir el K donde la tasa de error comienza a disminuir a un ritmo más lento, lo que indica un equilibrio entre sesgo y varianza.

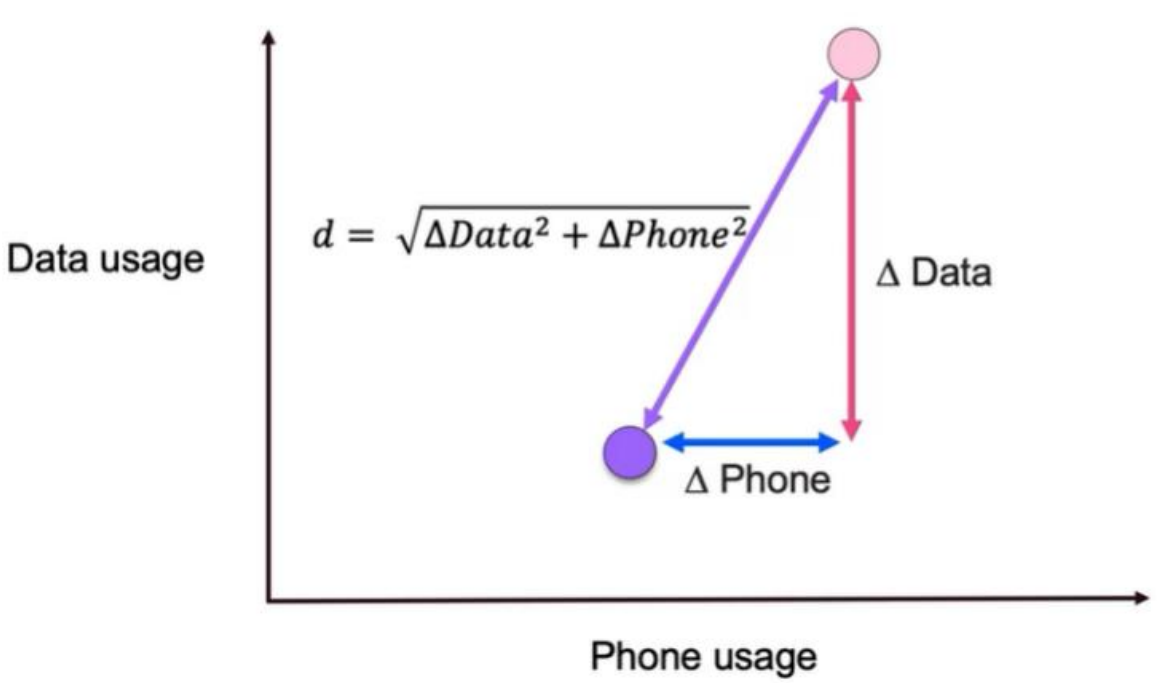
2. Midiendo la distancia



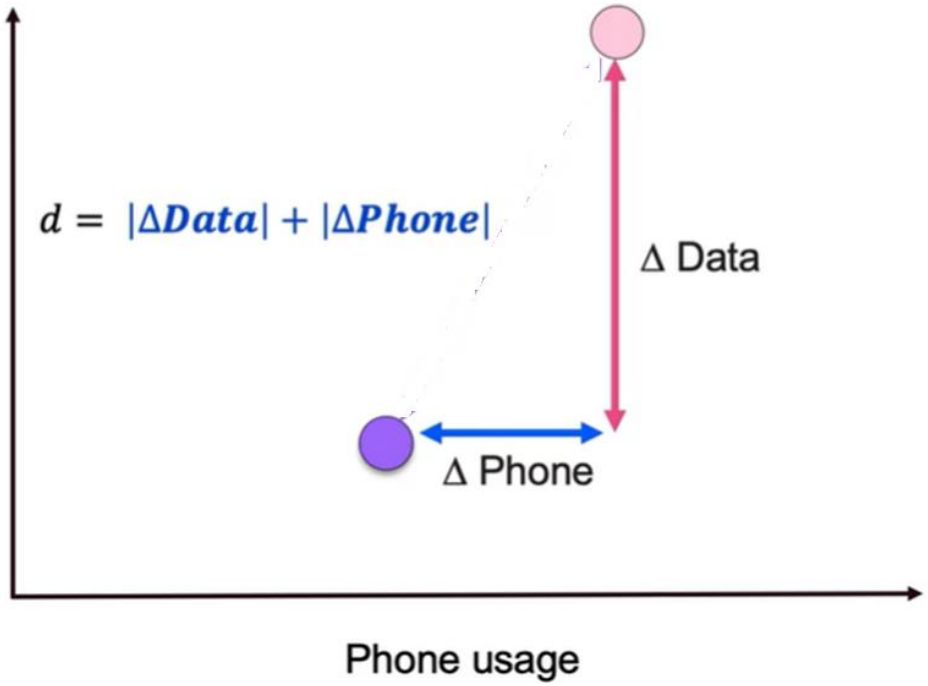
Cada punto tiene una ubicación en el espacio, si tenemos 2 variables independientes(X) podemos graficarlo en un plano cartesiano, si fueran 3 variables sería como en la imagen de al lado.

2. Midiendo la distancia

Euclidean Distance



Manhattan Distance



2. Midiendo la distancia

Minkowsky:

$$D(\boldsymbol{x},\boldsymbol{y}) = \left(\sum_{i=1}^{m} |x_i - y_i|^r\right)^{1/r}$$

Euclidean:

$$D(\boldsymbol{x},\boldsymbol{y}) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2}$$

Manhattan / city-block:

$$D(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{m} |x_i - y_i|$$

Camberra:

$$D(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{m} \frac{|x_i - y_i|}{|x_i + y_i|}$$

Chebychev:

$$D(\boldsymbol{x},\boldsymbol{y}) = \max_{i=1}^{m} |x_i - y_i|$$

Quadratic:

$$D(\boldsymbol{x},\boldsymbol{y}) = (\boldsymbol{x} - \boldsymbol{y})^T Q(\boldsymbol{x} - \boldsymbol{y}) = \sum_{j=1}^{m}\left(\sum_{i=1}^{m}(x_i - y_i)q_{ji}\right)(x_j - y_j)$$

Q is a problem-specific positive definite $m \times m$ weight matrix

Mahalanobis:

$$D(\boldsymbol{x},\boldsymbol{y}) = [\det V]^{1/m}(\boldsymbol{x} - \boldsymbol{y})^{\mathrm{T}} V^{-1}(\boldsymbol{x} - \boldsymbol{y})$$

V is the covariance matrix of $A_1..A_m$, and A_j is the vector of values for attribute j occuring in the training set instances $1..n$.

Correlation:

$$D(\boldsymbol{x},\boldsymbol{y}) = \frac{\sum_{i=1}^{m}(x_i - \overline{x_i})(y_i - \overline{y_i})}{\sqrt{\sum_{i=1}^{m}(x_i - \overline{x_i})^2 \sum_{i=1}^{m}(y_i - \overline{y_i})^2}}$$

$\overline{x_i} = \overline{y_i}$ and is the average value for attribute i occuring in the training set.

Chi-square:

$$D(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{m} \frac{1}{sum_i}\left(\frac{x_i}{size_{\boldsymbol{x}}} - \frac{y_i}{size_{\boldsymbol{y}}}\right)^2$$

sum_i is the sum of all values for attribute i occuring in the training set, and $size_{\boldsymbol{x}}$ is the sum of all values in the vector $\boldsymbol{x}$.

Kendall's Rank Correlation:

$$D(\boldsymbol{x},\boldsymbol{y}) = 1 - \frac{2}{n(n-1)}\sum_{i=1}^{m}\sum_{j=1}^{i-1}\mathrm{sign}(x_i - x_j)\mathrm{sign}(y_i - y_j)$$

sign(x)=-1, 0 or 1 if $x < 0$, $x = 0$, or $x > 0$, respectively.

Figure 1. Equations of selected distance functions.
($\boldsymbol{x}$ and $\boldsymbol{y}$ are vectors of m attribute values).

3. Pesos

Uniforme (por defecto)

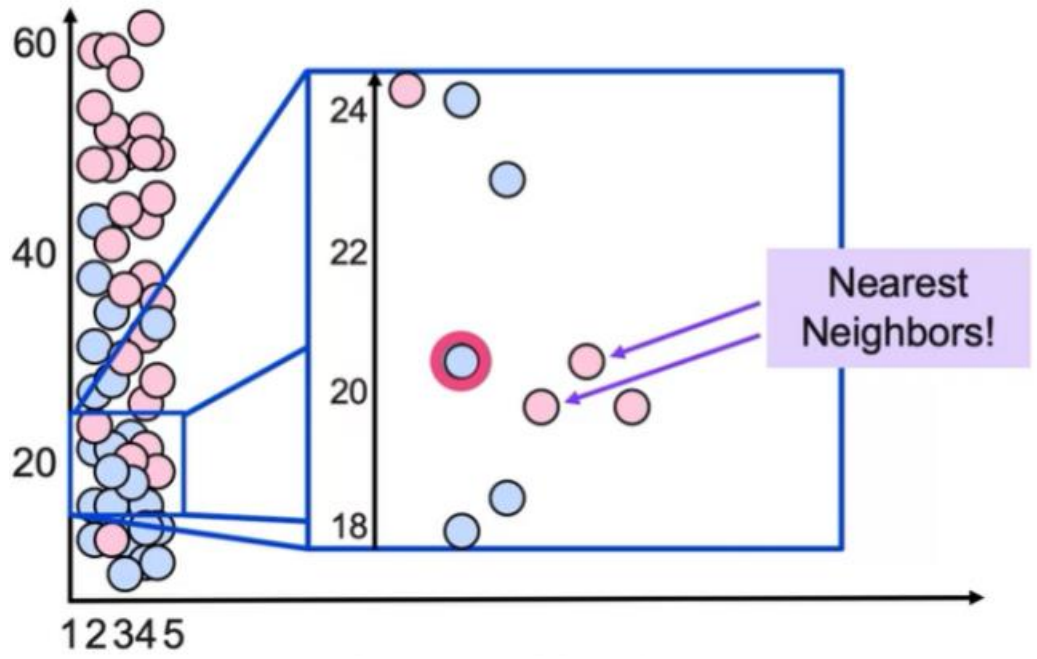
Todos los vecinos más cercanos tienen el **mismo peso en la decisión final**. En otras palabras, la clase que tenga la mayor cantidad de representantes dentro de los k vecinos más cercanos será la clase predicha.

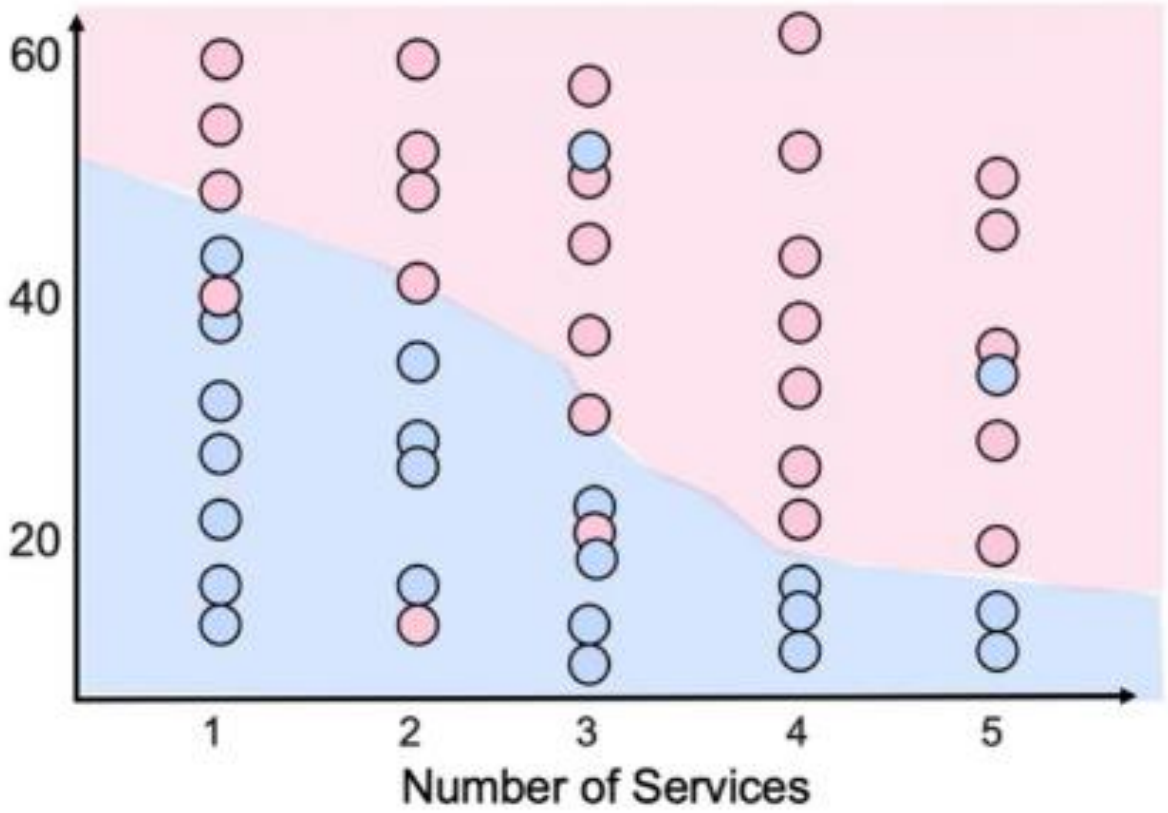
Utiliza k impar

Distance

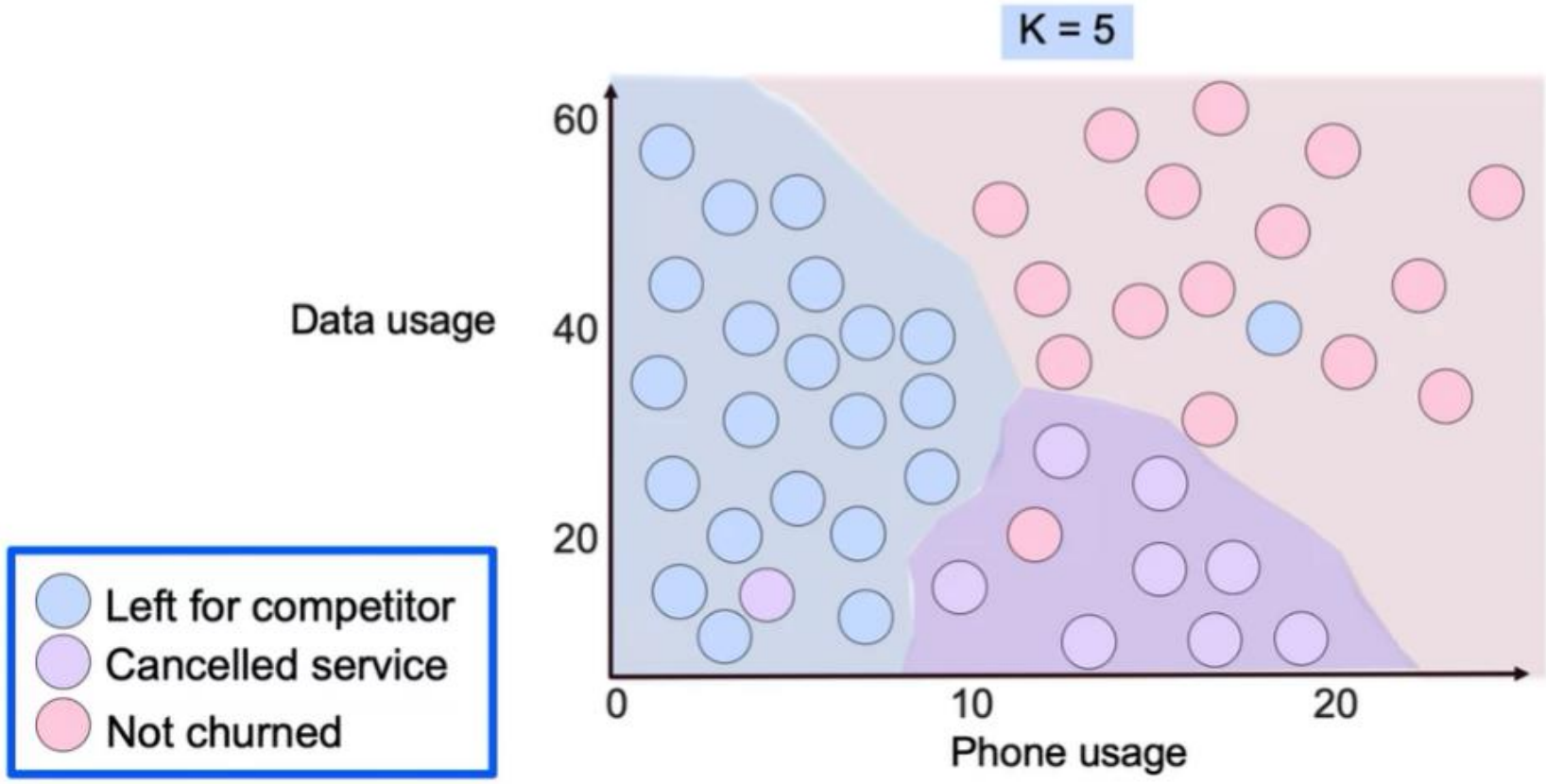
Los vecinos más cercanos tienen un peso mayor en la votación o en la predicción. Específicamente, **el peso de cada vecino es inversamente proporcional a su distancia** del punto de consulta

Escalar es esencial!



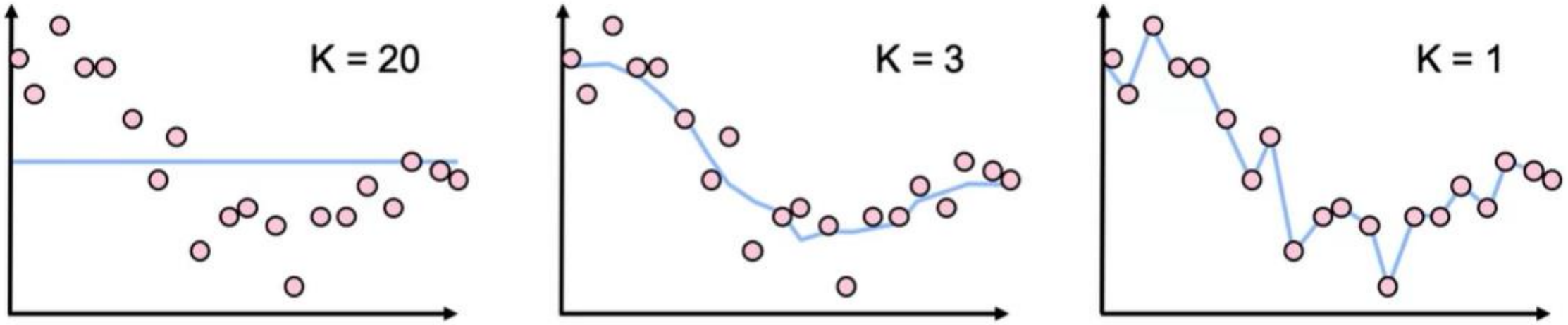


No sobra decir, que puede tener múltiples clases



Regresión

En la regresión KNN, el valor predicho es el valor medio de los K vecinos más cercanos. Esto permite a KNN predecir valores continuos, similar a su enfoque de clasificación pero centrándose en el promedio de los vecinos.



K Nearest Neighbors: The Syntax

Code

```
# Import the class containing the classification method
from sklearn.neighbors import KNeighborsClassifier

# Create an instance of the class
KNN = KNeighborsClassifier(n_neighbors=3)

# Fit the instance on the data and then predict the expected value
KNN = KNN.fit(X_train, y_train)
y_predict = KNN.predict(X_test)

Regression can be done with KNeighborsRegressor.
```

	Ventajas	Desventajas
	Entrena rápido	Predición lenta por el calculo de tantas distancias
	Simple, adaptable	Intensivo en memoria
	Alta exactitud	Maldición de la dimensionalidad