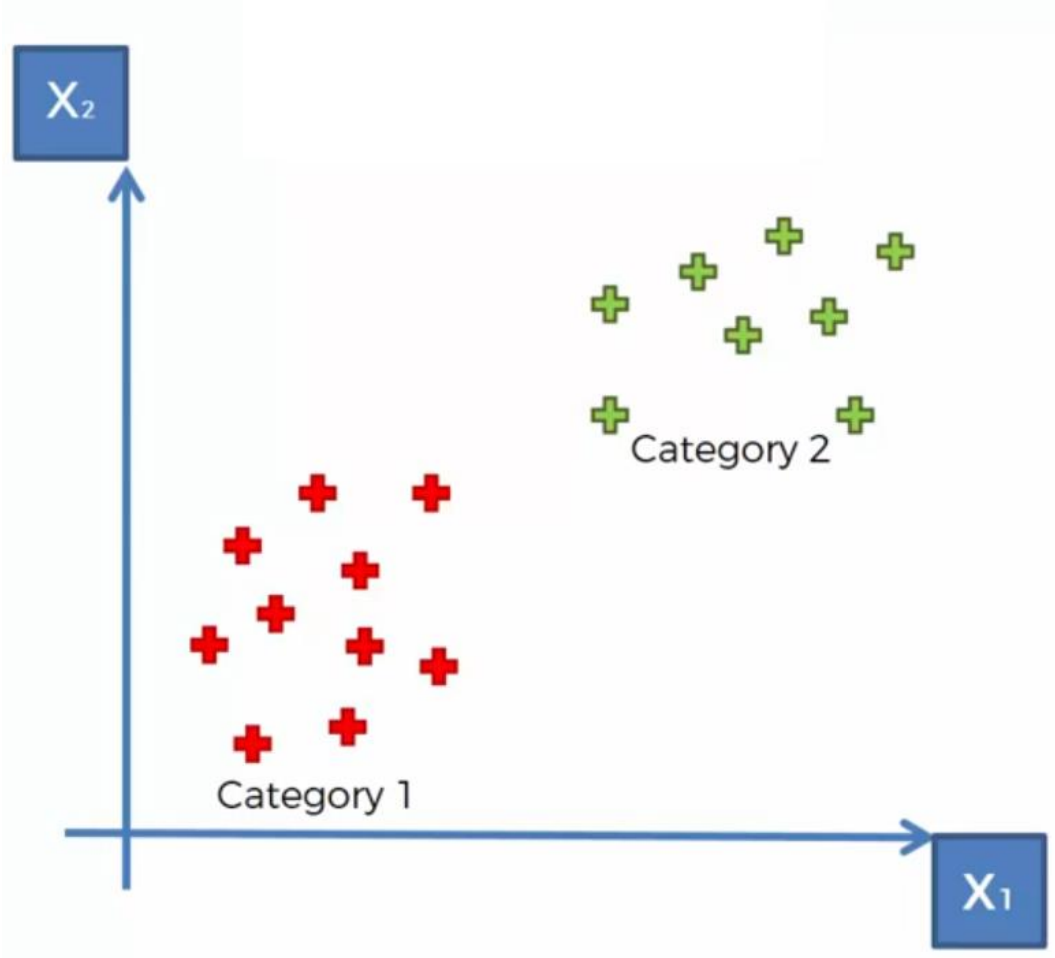
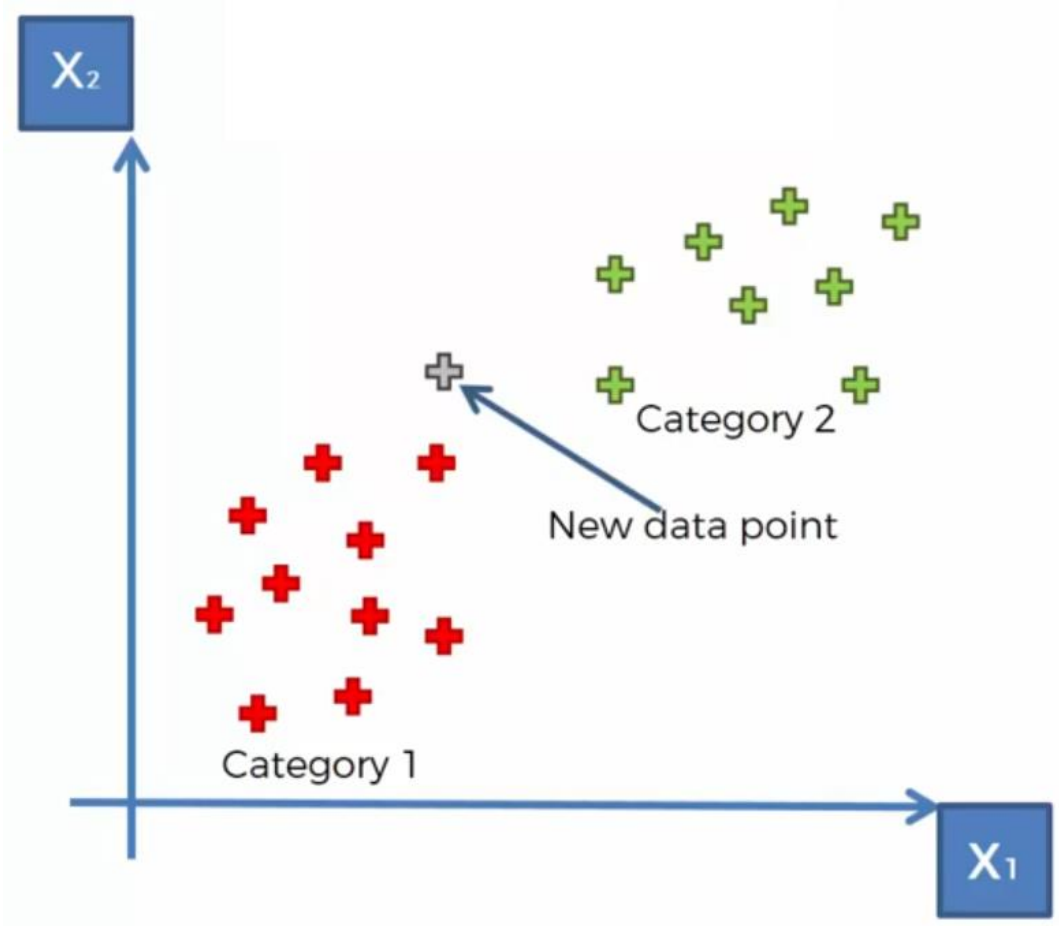
Introduction to Machine Learning. Lec.10 K-Nearest Neighbors

Aidos Sarsembayev, IITU, 2018

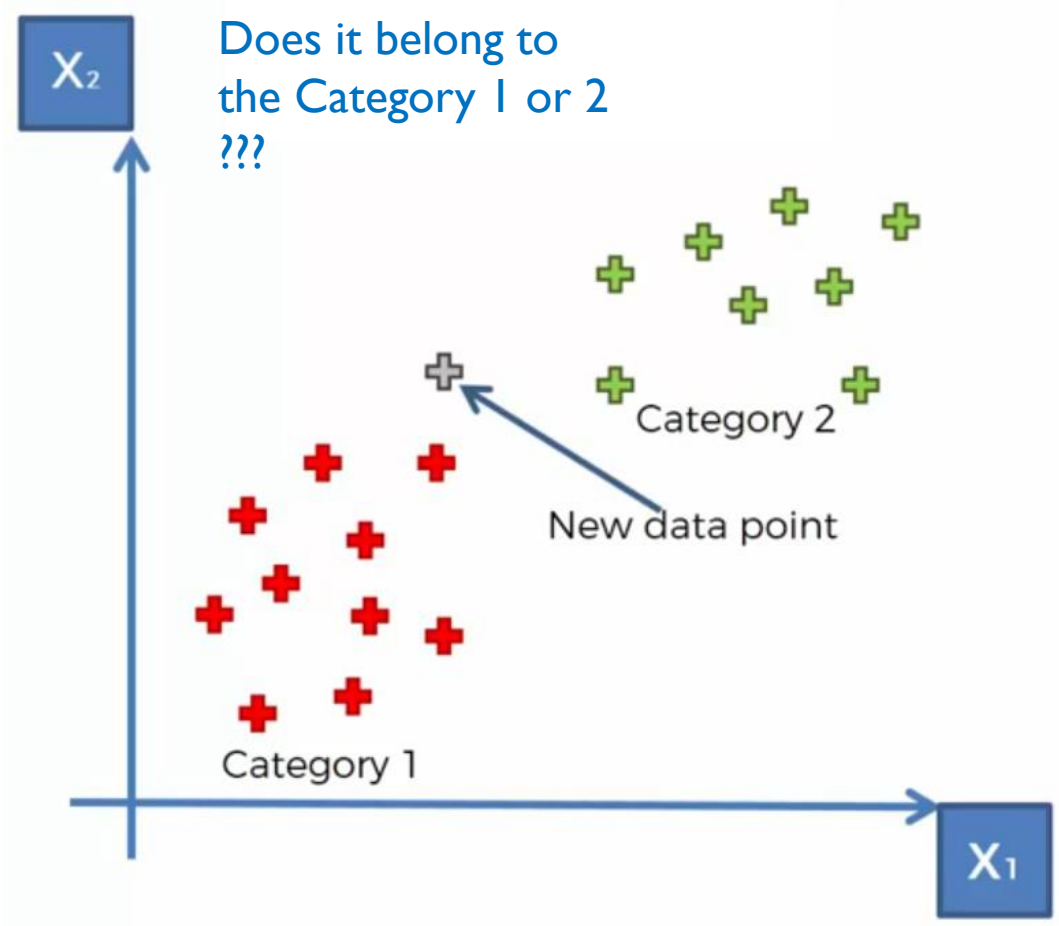
Before k-NN



A new data point



A new data point



After k-NN applied

k-NN algorithm step-by-step

- STEP 1: Choose the number K of neighbors

k-NN algorithm step-by-step

- STEP 1: Choose the number k of neighbors
  Most commonly k=5 (it's better to choose odd numbers)

k-NN algorithm step-by-step

- STEP 1: Choose the number K of neighbors
- STEP 2: Take the k nearest neighbors of the new data point, according to the Euclidean Distance.

k-NN algorithm step-by-step

- STEP 1: Choose the number K of neighbors
- STEP 2: Take the k nearest neighbors of the new data point, according to the Euclidean Distance.

$$d(p_1, p_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \text{ where } p_1 = (x_1, y_1) \text{ and } p_2 = (x_2, y_2)$$

k-NN algorithm step-by-step

- STEP 1: Choose the number K of neighbors
- STEP 2: Take the k nearest neighbors of the new data point, according to the Euclidean Distance.

$$d(p_1, p_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \text{ where } p_1 = (x_1, y_1) \text{ and } p_2 = (x_2, y_2)$$

you don't strictly have to use Euclidean distance, but it's the most common one

k-NN algorithm step-by-step

- STEP 1: Choose the number K of neighbors
- STEP 2: Take the k nearest neighbors of the new data point, according to the Euclidean Distance.
- STEP 3: Among these k neighbors, count the number of data points in each category

k-NN algorithm step-by-step

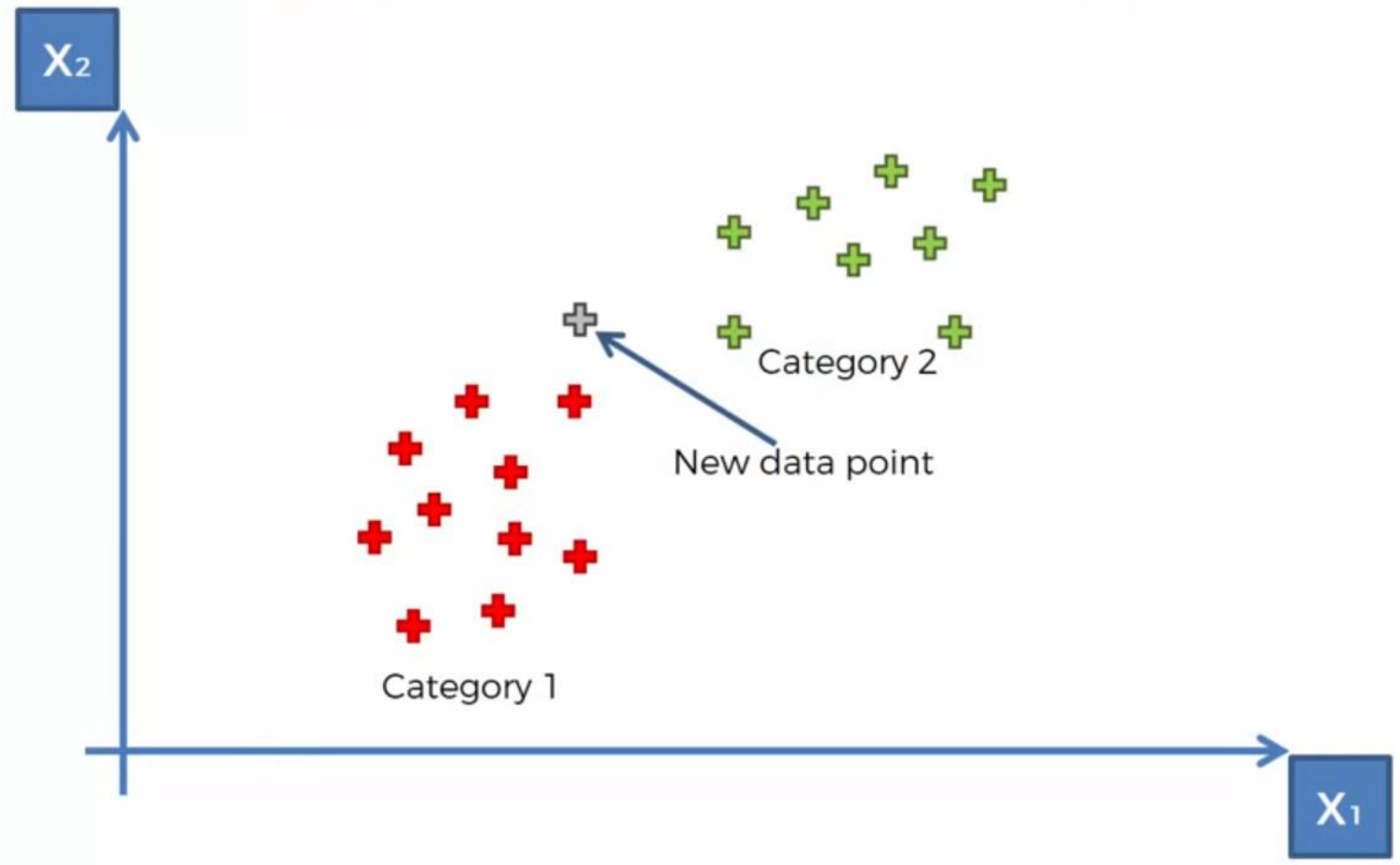
- STEP 1: Choose the number K of neighbors
- STEP 2: Take the k nearest neighbors of the new data point, according to the Euclidean Distance.
- STEP 3: Among these k neighbors, count the number of data points in each category
- STEP 4: Assign the new data point to the category where you counted the most neighbors

STEP 1: Choose the number K of neighbors (K=5)

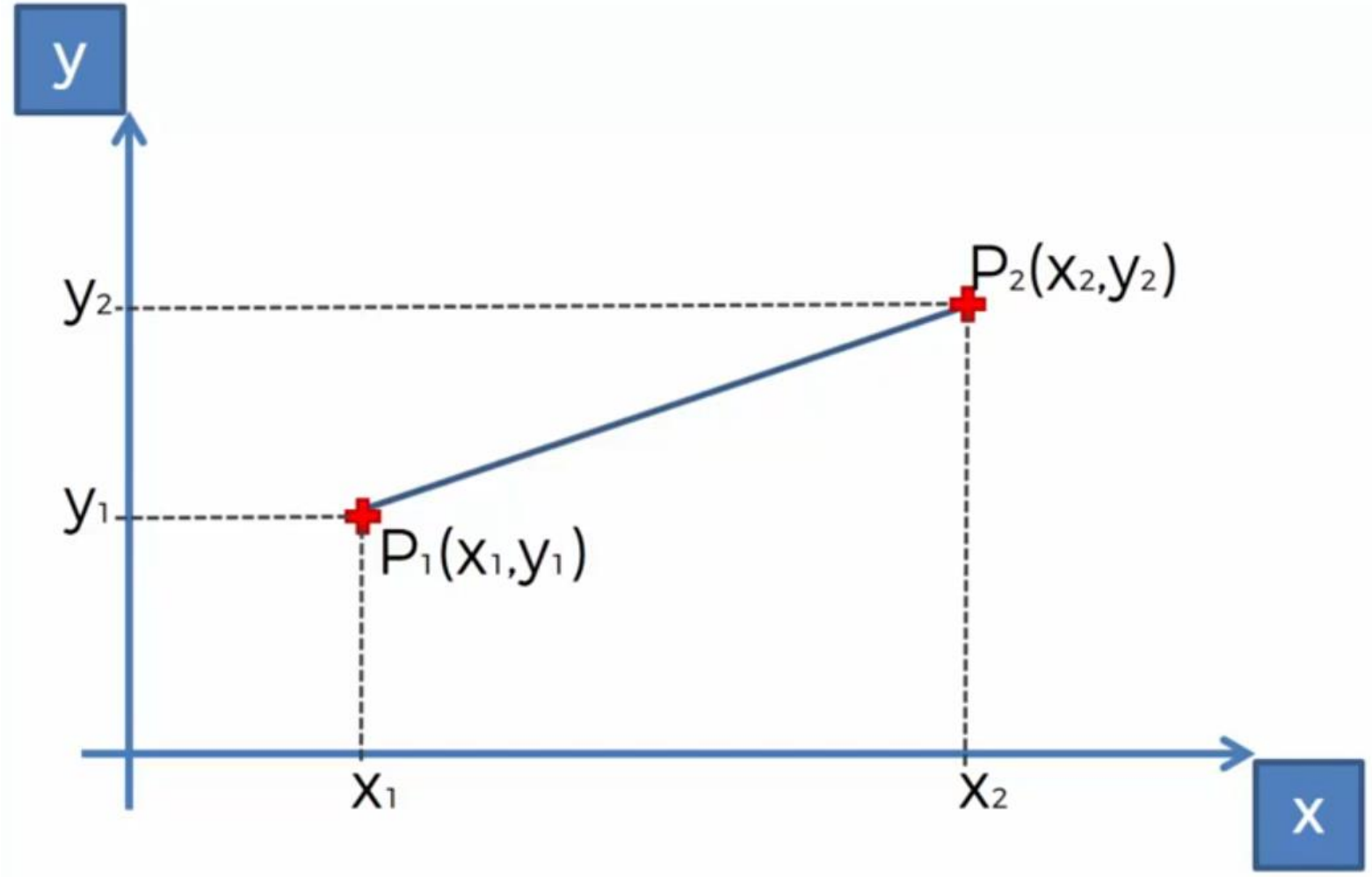
How to choose k?


K-NEAREST NEIGHBORS
K = 3
risovach.ru

STEP 2: Take the k nearest neighbors of the new data point, according to the Euclidean Distance.



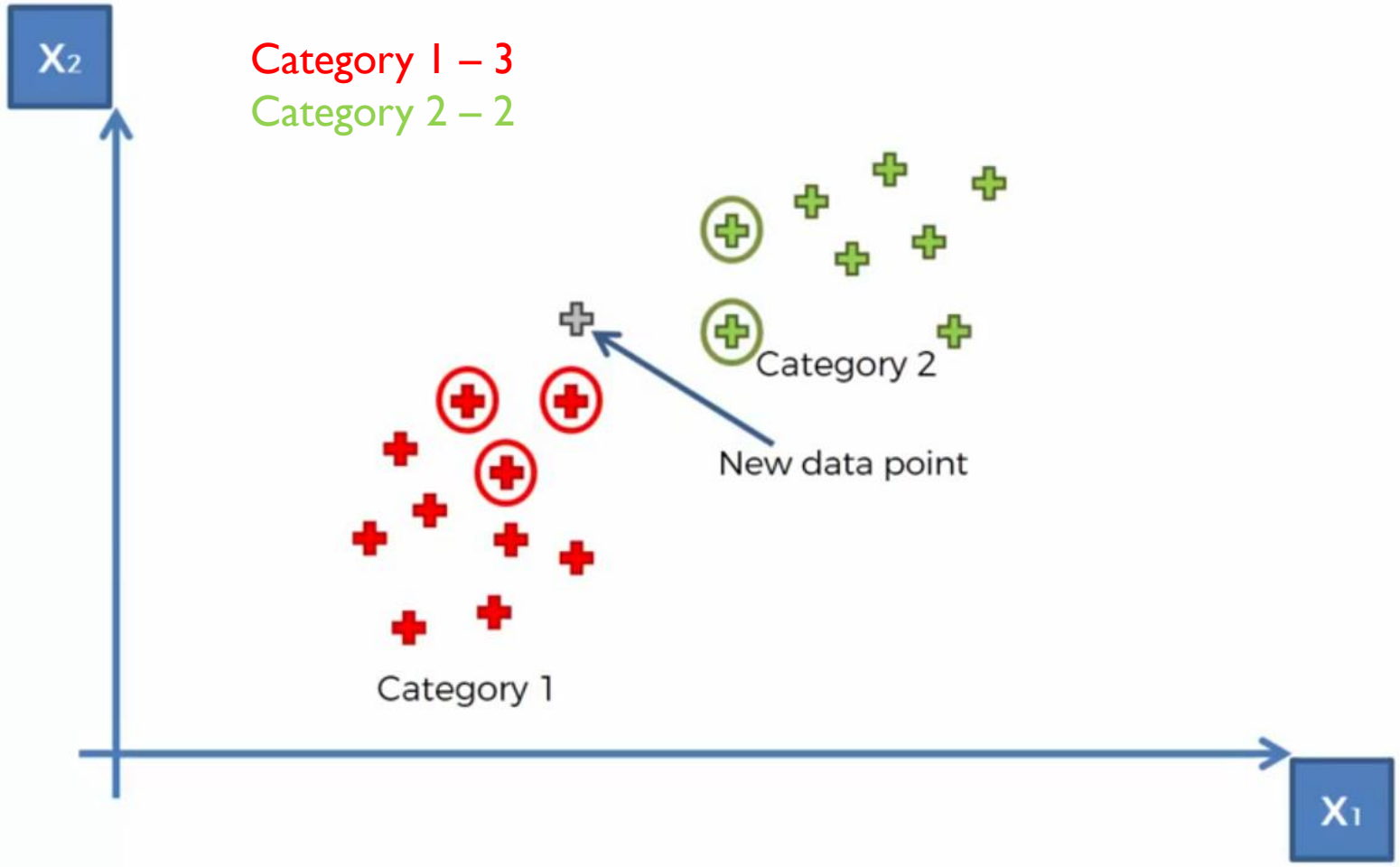
STEP 2: Take the k nearest neighbors of the new data point, according to the Euclidean Distance.



STEP 2: Take the k nearest neighbors of the new data point, according to the Euclidean Distance.

STEP 3: Among these k neighbors, count the number of data points in each category

STEP 3: Among these k neighbors, count the number of data points in each category



STEP 4: Assign the new data point to the category where you counted the most neighbors

Class creation

- sklearn.neighbors -> KNeighborsClassifier

Class creation

- sklearn.neighbors -> KNeighborsClassifier
- Parameters:

Class creation

- sklearn.neighbors -> KNeighborsClassifier
- Parameters: n_neighbors (i.g. n_neighbors= 5)

Class creation

- sklearn.neighbors -> KNeighborsClassifier
- Parameters:
  n_neighbors (i.g. n_neighbors= 5)
  metric (i.g. metric = ‘minkowski’)

Class creation

- sklearn.neighbors -> KNeighborsClassifier
- Parameters:
  n_neighbors (i.g. n_neighbors= 5)
  metric (i.g. metric = ‘minkowski’)
  p (i.g. p =2 is Euclidean distance)

This is it ;-)