K-Nearest Neighbours (KNN)

Non Parametric, Lazy, Multi Class Classification Technique

K-NN

KNN algorithm is one of the simplest classification algorithm and it is one of the most used learning algorithms.

KNN can be used for classification—the output is a class membership (predicts a class—a discrete value). An object is classified by a majority vote of its ‘k’ neighbors, with the object being assigned to the class most common among its k nearest neighbors.

Features

KNN stores the entire training dataset which it uses as its representation.

KNN does not learn any model.

KNN makes predictions just-in-time by calculating the similarity between an input sample and each training instance.

Lazy

KNN is a **lazy** algorithm (as opposed to an *eager* algorithm). This means it does not use the training data points to do any *generalization*.

In other words, there is *no explicit training phase* or it is very minimal. This also means that the training phase is pretty fast . Lack of generalization means that KNN keeps all the training data.

To be more exact, all (or most) the training data is needed during the testing phase.

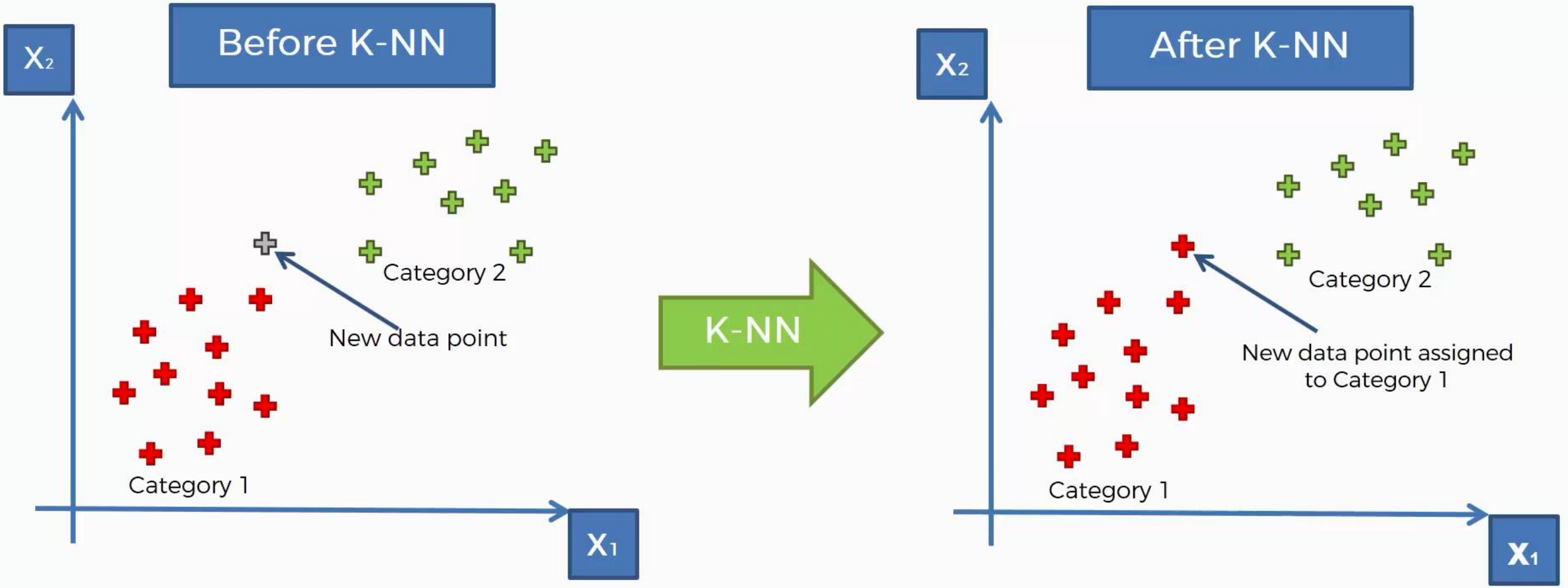
Non-Parametric

KNN does not make any assumptions on the underlying data distribution. In other words, the model structure is determined from the data.

If you think about it, it's pretty useful, because in the "real world", most of the data does not obey the typical theoretical assumptions made (as in linear regression models, for example).

Therefore, KNN could and probably should be one of the first choices for a classification study when there is little or no prior knowledge about the distribution data.

What K-NN does for you



How did it do that ?

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

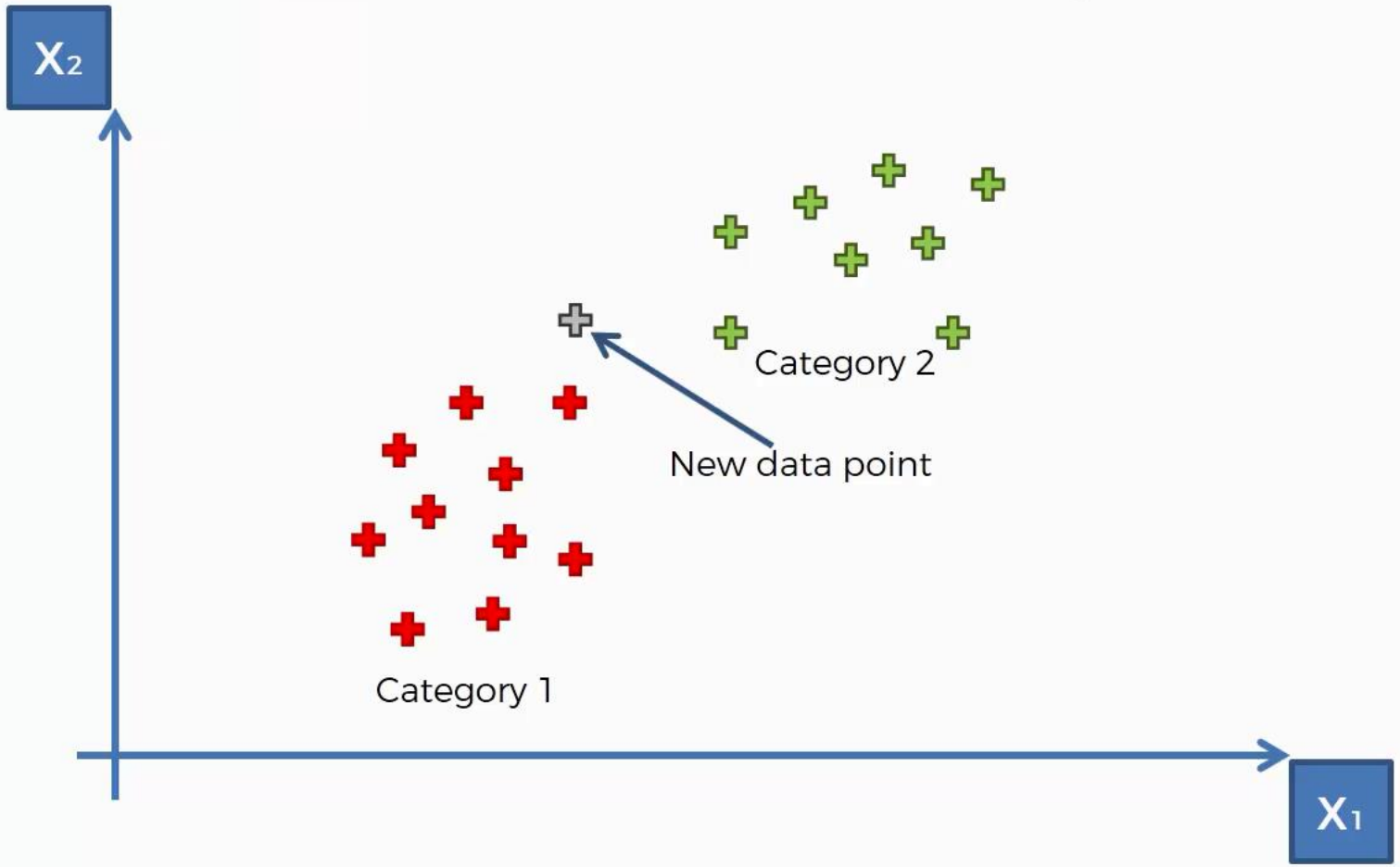
STEP 3: Among these K neighbors, count the number of data points in each category

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

Your Model is Ready

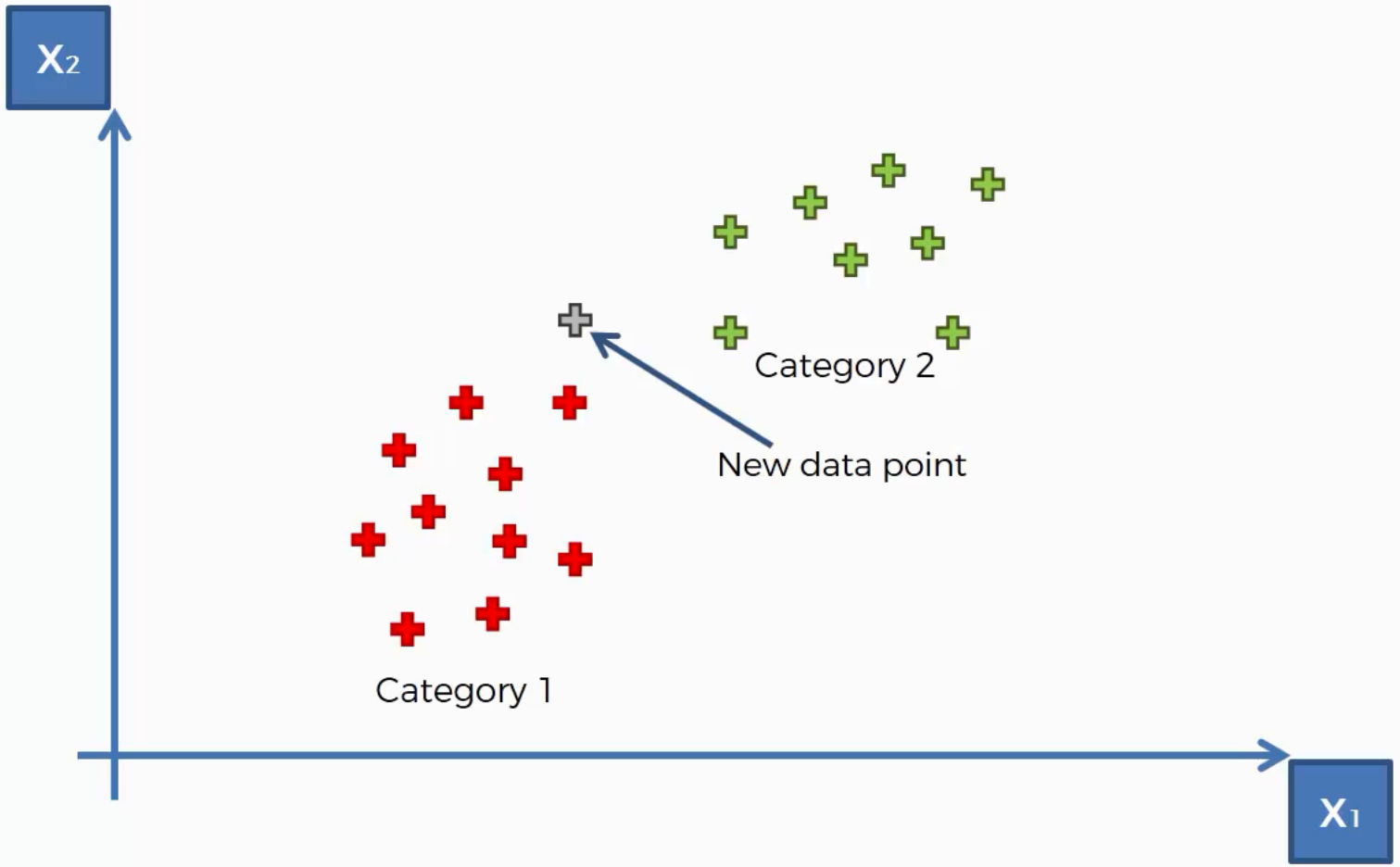
K-NN algorithm

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

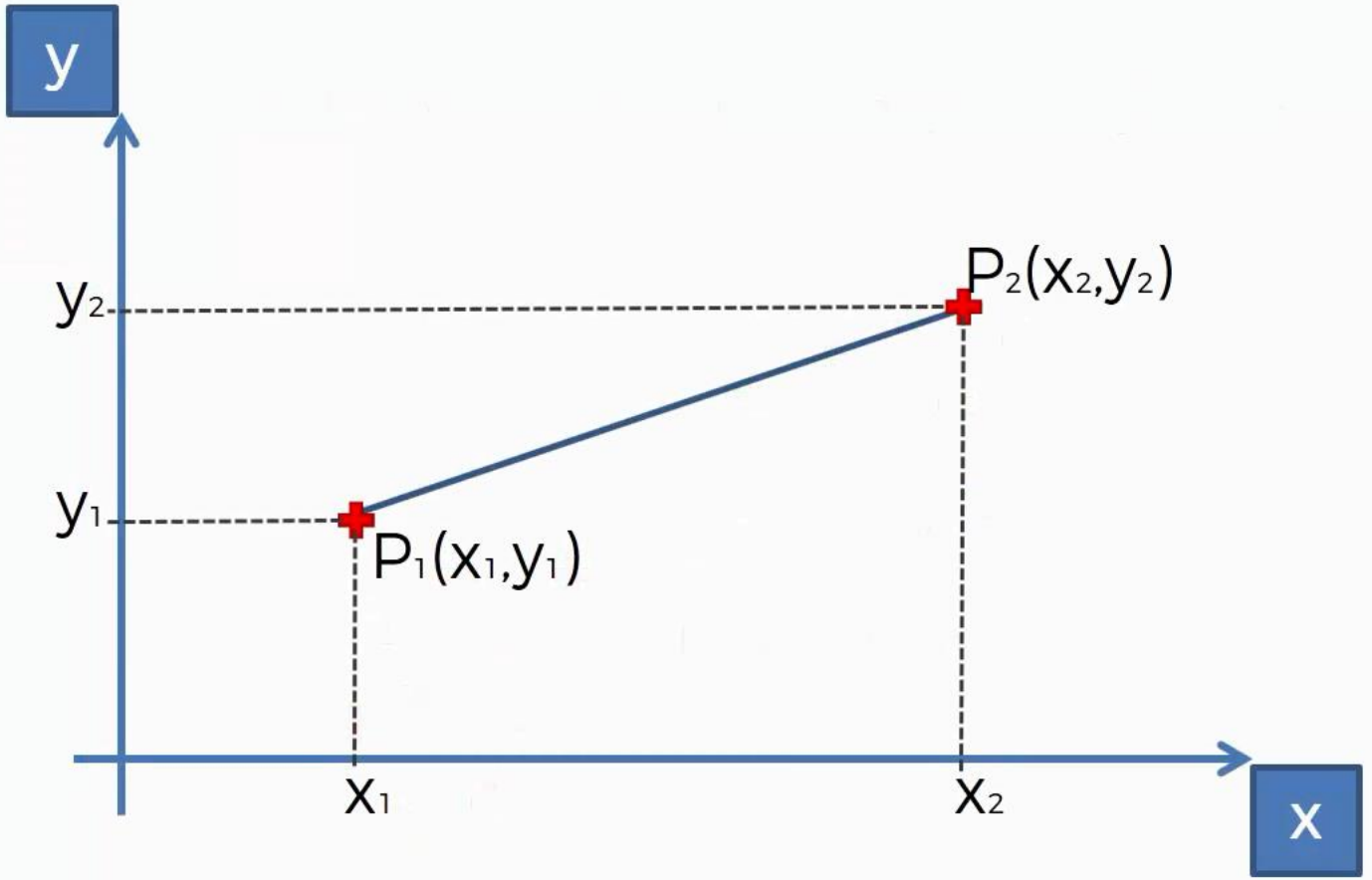


K-NN algorithm

STEP 2: Take the K = 5 nearest neighbors of the new data point, according to the Euclidean distance



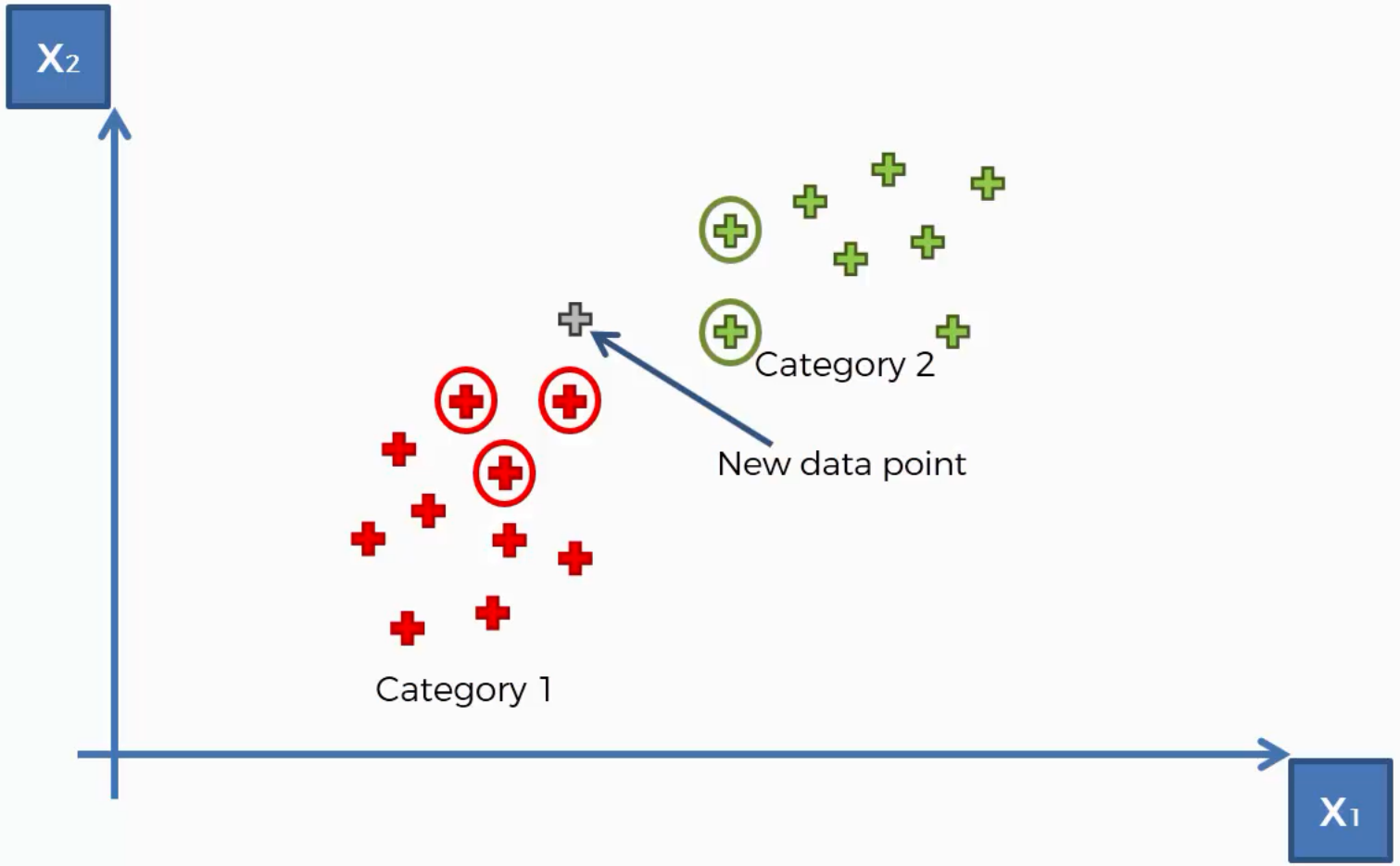
Euclidean Distance



Euclidean Distance between P_1 and $P_2 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

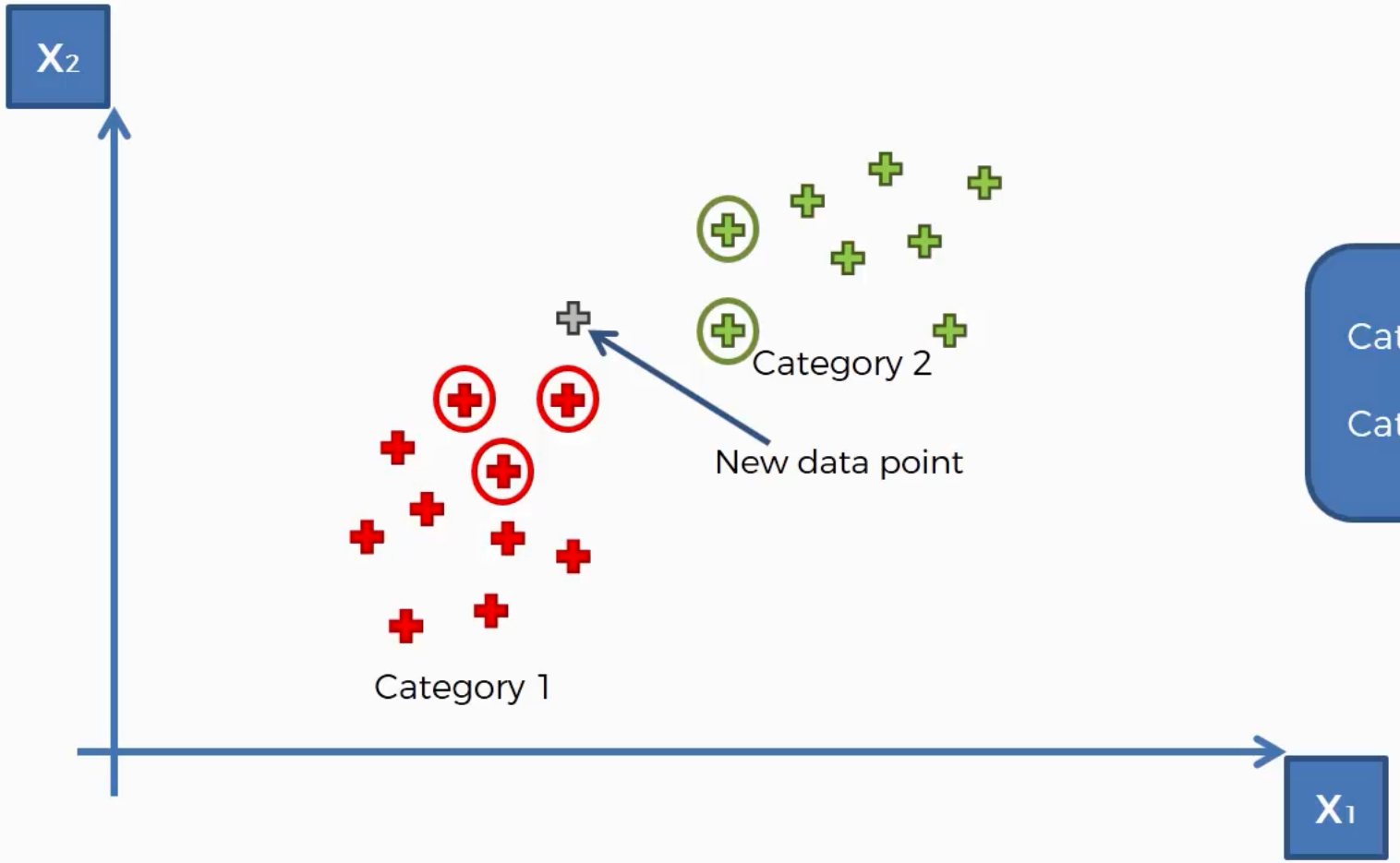
K-NN algorithm

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



K-NN algorithm

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

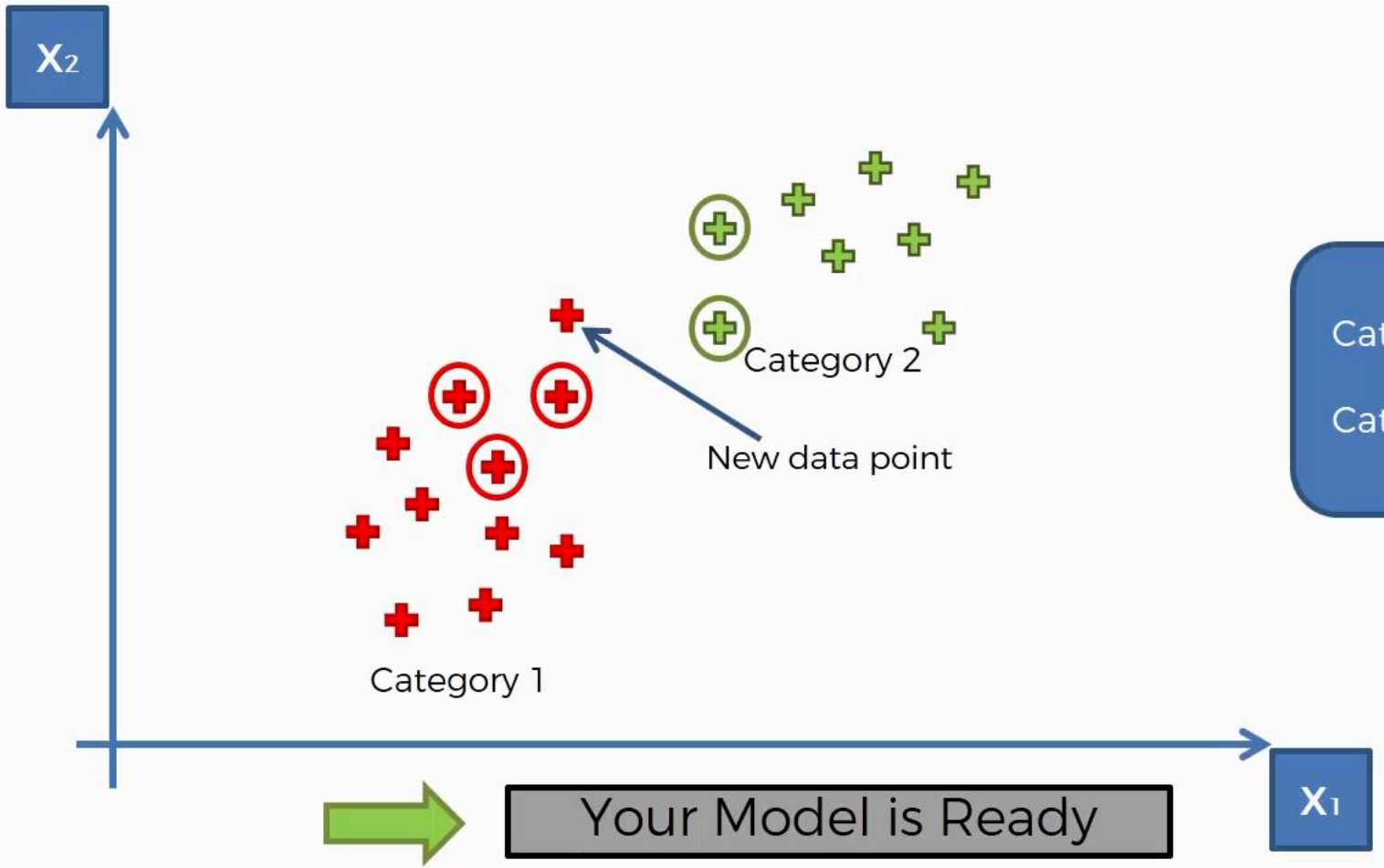


Category 1: 3 neighbors

Category 2: 2 neighbors

K-NN algorithm

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



Category 1: 3 neighbors

Category 2: 2 neighbors

Your Model is Ready

Ways to find Distances:

Euclidean

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

Manhattan

$$\sum_{i=1}^{k}|x_i - y_i|$$

Minkowski

$$\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$$

It should also be noted that all three distance measures are only valid for continuous variables.

In the instance of categorical variables the Hamming distance must be used.

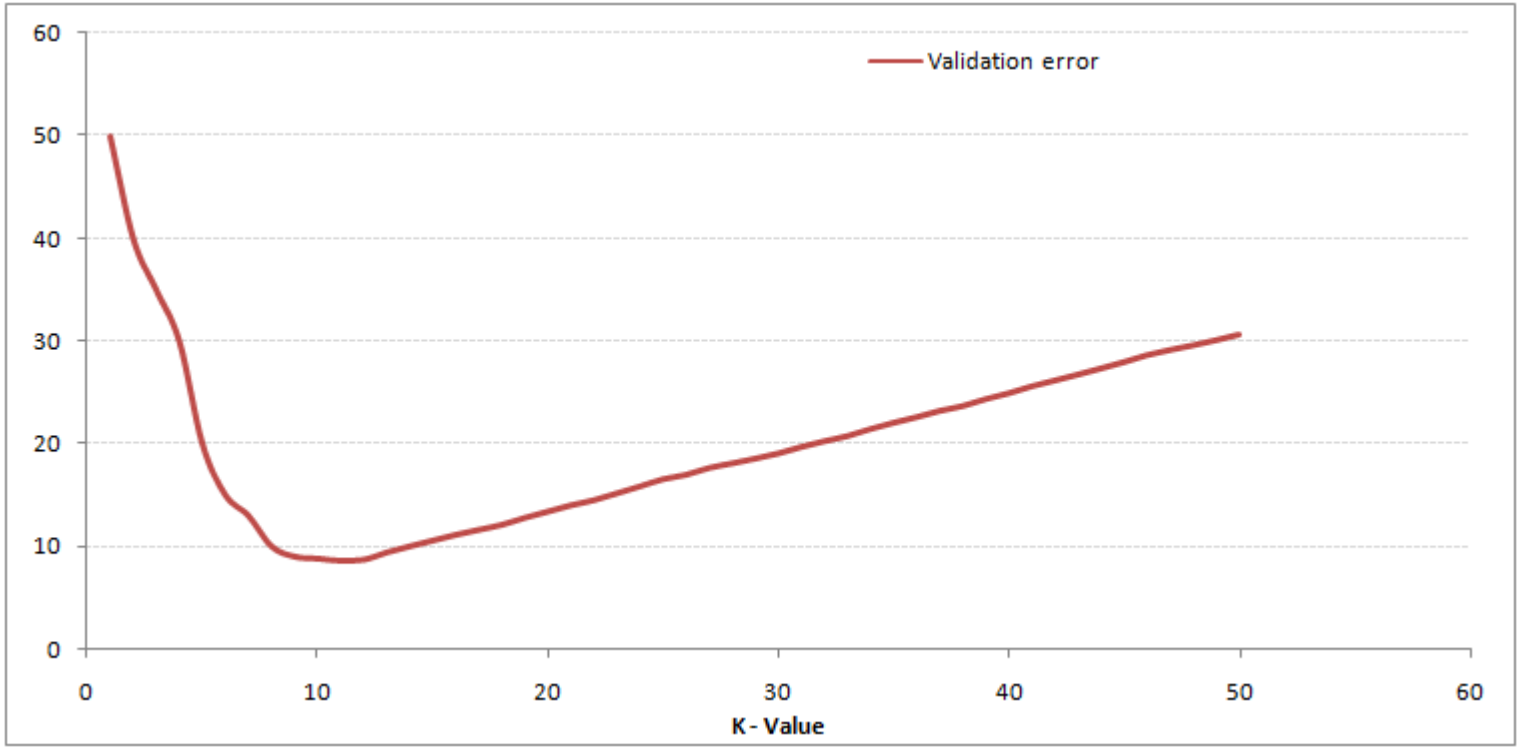
It also brings up the issue of standardization of the numerical variables between 0 and 1 when there is a mixture of numerical and categorical variables in the dataset.

K? How many?

Choosing the optimal value for K is best done by first inspecting the data. In general, a large K value is more precise as it reduces the overall noise but there is no guarantee.

Cross-validation is another way to retrospectively determine a good K value by using an independent dataset to validate the K value.

Historically, the optimal K for most datasets has been between 3-10. That produces much better results than 1NN.



This makes the story more clear. At K=1, we were overfitting the boundaries. Hence, error rate initially decreases and reaches a minima. After the minima point, it then increase with increasing K. To get the optimal value of K, you can segregate the training and validation from the initial dataset. Now plot the validation error curve to get the optimal value of K. This value of K should be used for all predictions.

Standardized Distances

One major drawback in calculating distance measures directly from the training set is when variables have different measurement scales or there is a mixture of numerical and categorical variables.

For example, if one variable is based on annual income in dollars, and the other is based on age in years then income will have a much higher influence on the distance calculated. One solution is to standardize the training set.

Do Feature Scaling before making the K-NN model.

How to handle categorical variables in KNN?

Create dummy variables out of a categorical variable and include them instead of original categorical variable.

Unlike regression, create k dummies instead of (k-1).

For example, a categorical variable named "Department" has 5 unique levels / categories. So we will create 5 dummy variables. Each dummy variable has 1 against its department and else 0.

Advantages

- Easy to understand
- No assumptions about data
- Can be applied to both classification and regression
- Works easily on multi-class problems

Disadvantages

- Computationally expensive — because the algorithm stores all of the training data
- High memory requirement - Stores all (or almost all) of the training data
- Does not work well with rare erroneous events in the data
- Struggles with high number of independent variables

Applications & Examples

- Credit ratings—collecting financial characteristics vs. comparing people with similar financial features to a database. By the very nature of a credit rating, people who have similar financial details would be given similar credit ratings. Therefore, they would like to be able to use this existing database to predict a new customer's credit rating, without having to perform all the calculations.
- Should the bank give a loan to an individual? Would an individual default on his or her loan? Is that person closer in characteristics to people who defaulted or did not default on their loans?
- Handwriting detection (like OCR), image recognition and even video recognition.

Can K-NN be used for Regression?

Yes, K-nearest neighbor can be used for regression. In other words, K-nearest neighbor algorithm can be applied when dependent variable is continuous.

In this case, the predicted value is the average of the values of its k nearest neighbors.