Index Class

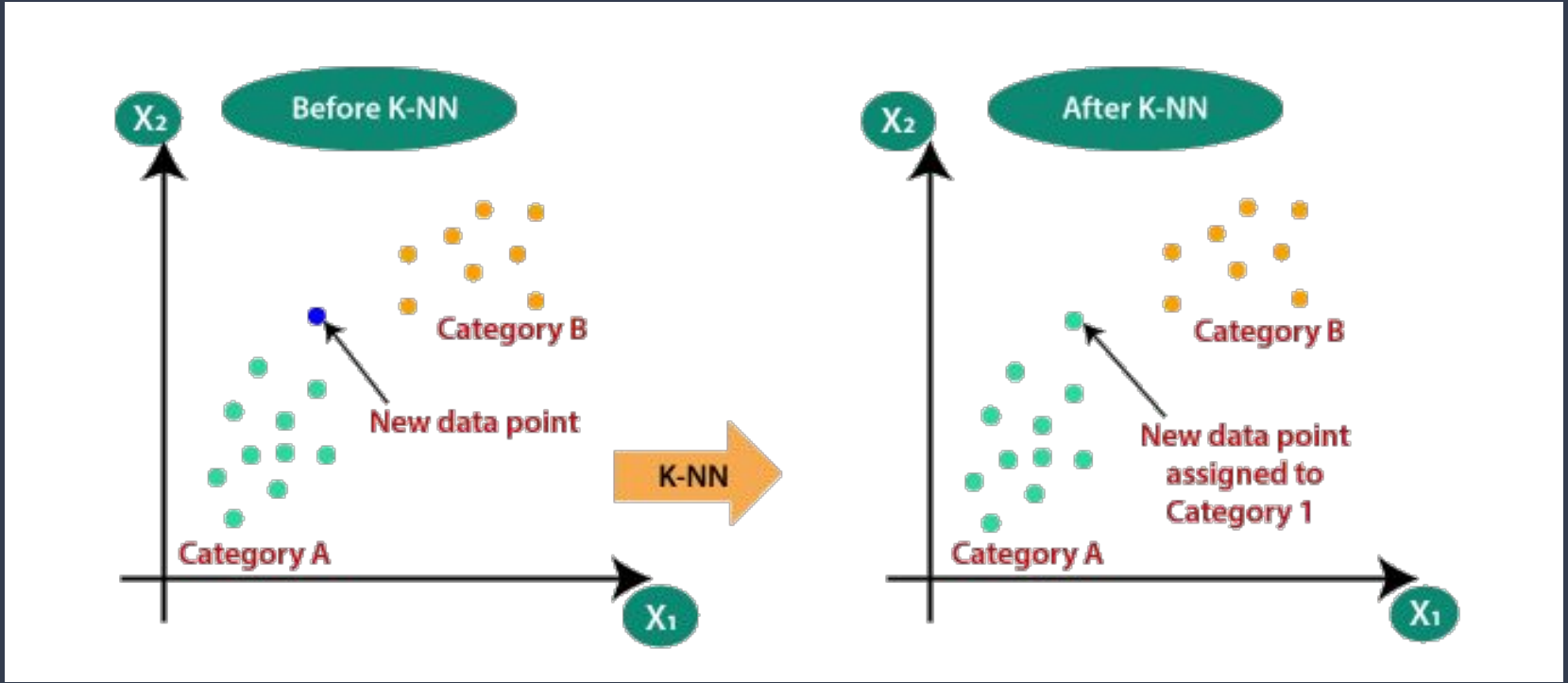
1. KNN
2. Problems and improvements
3. Overfitting and bias-variance trade off
4. From LR to KNN

K Nearest Neighbor

Supervised learning

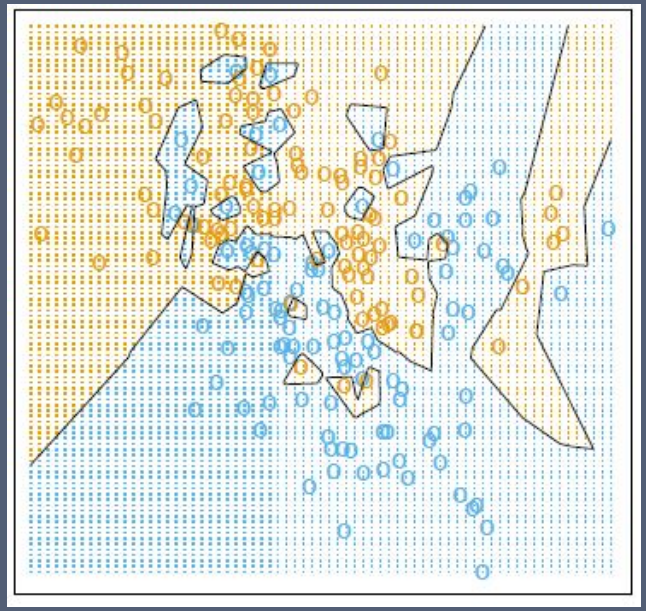
Non-parametric

K Nearest Neighbor

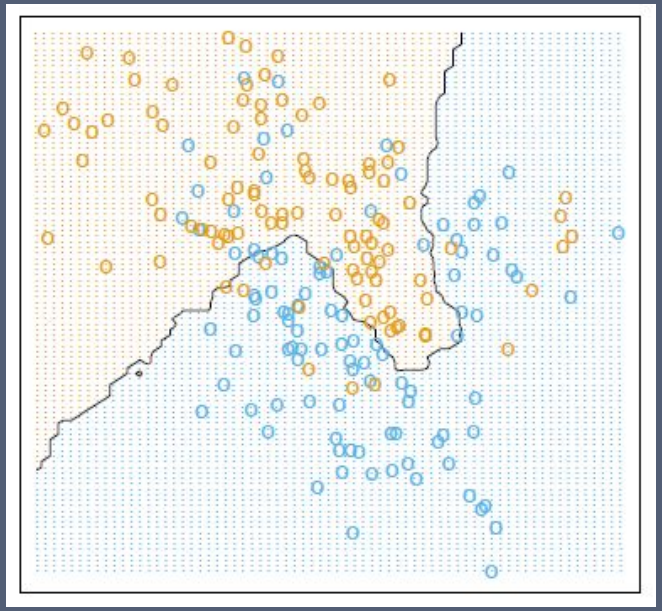


From parameters to hyper-parameters

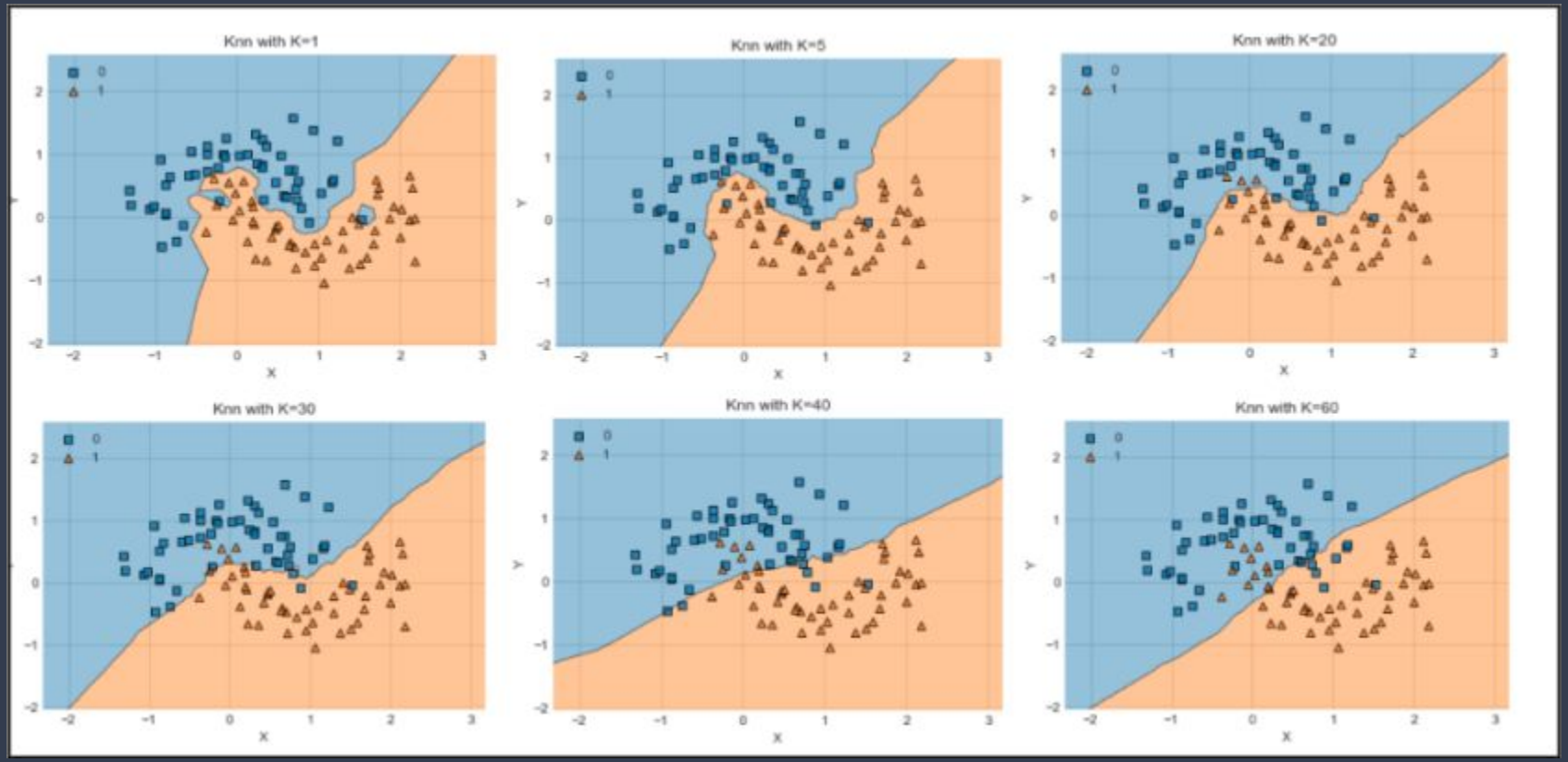
K = 1



K = 20



K Nearest Neighbor



K Nearest Neighbour

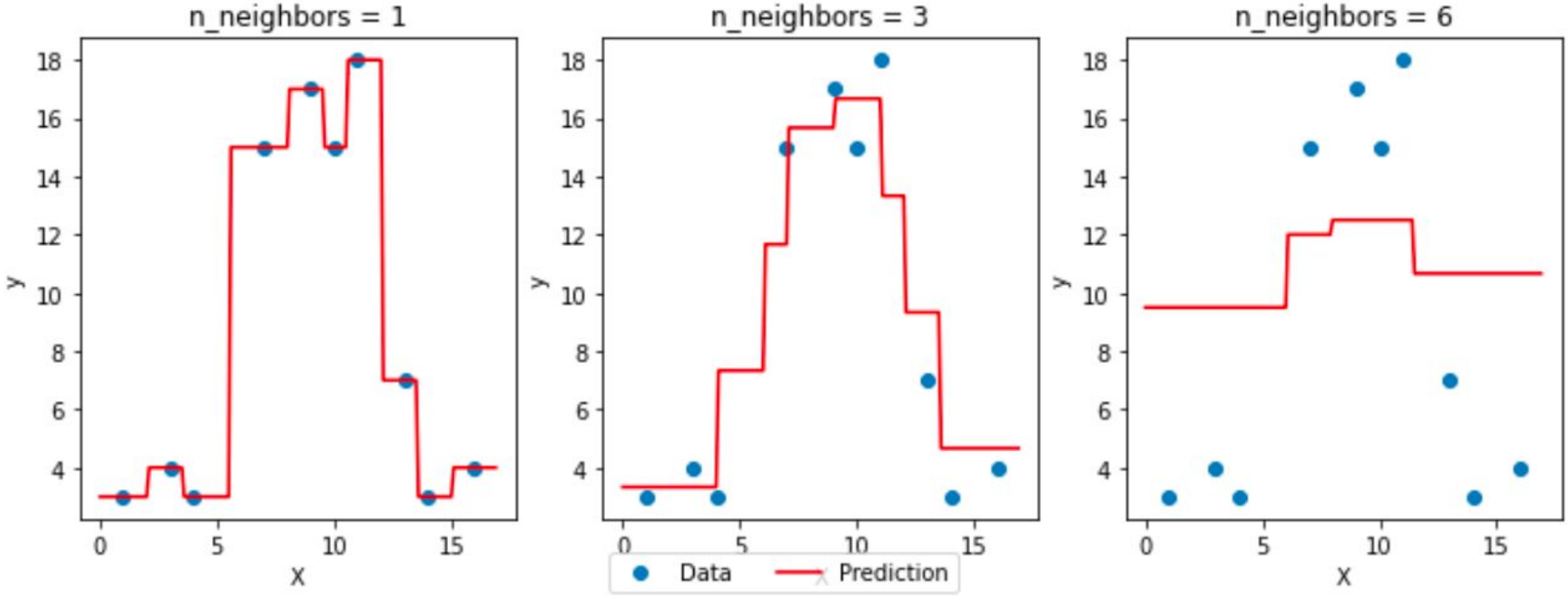
Nk(x) is the neighborhood of *x* defined by the *K* closest points *xi* in the training sample.

Closest implies a metric, we assume Euclidean distance, but any distance will be fine.

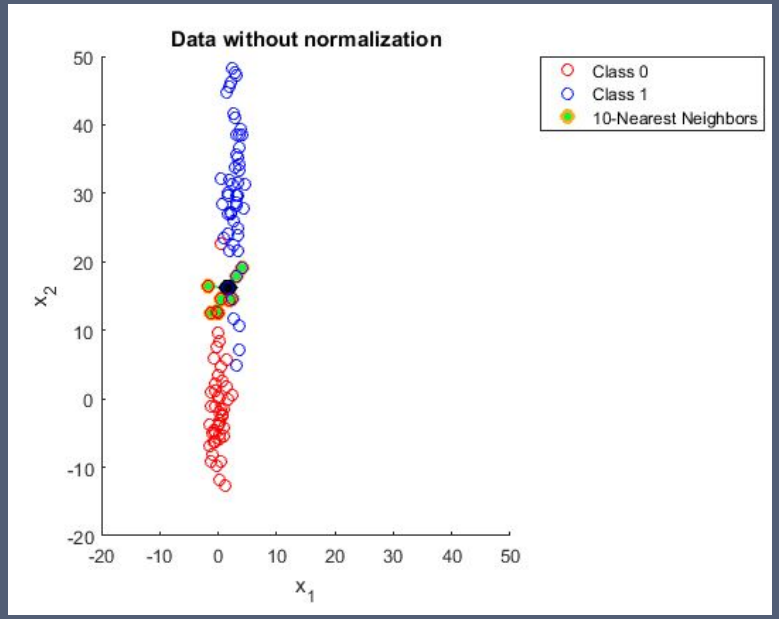
The response is the average response of the *K* closest observations (classification or regression)

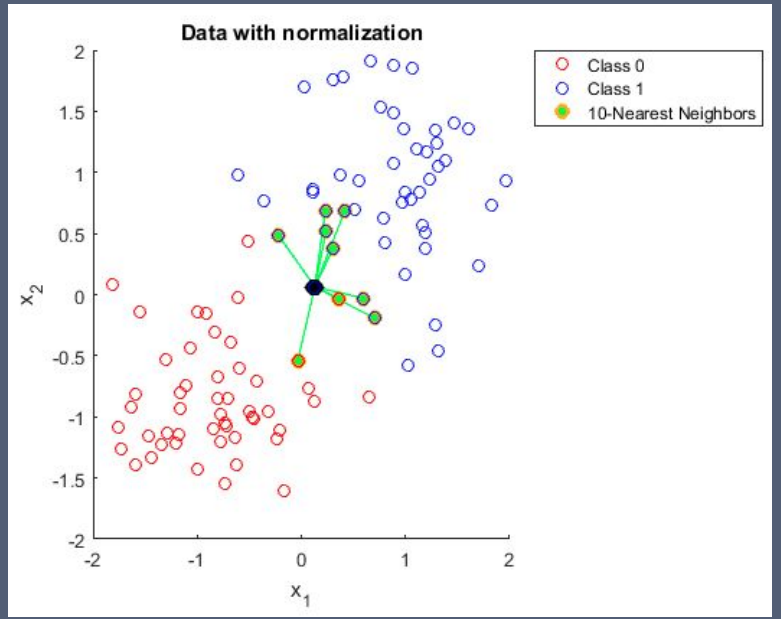
$$\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i$$

KNN Regressor



Problems in KNN: Distance





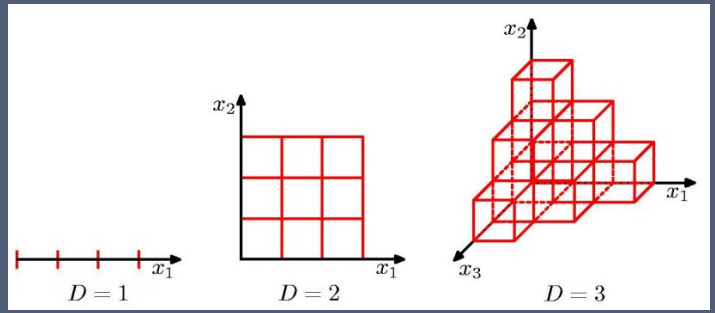
Problems in KNN: Many features

- High Dimensionality

Predictions get random as the number of features grow.

In the discrete case, number of neighbors increases:
1D: 2 - 4 - 6 - 8 - ...
2D: 4 - 12 - 20 - ...
3D: 6 - 18 - 32 - ...

It is highly recommended to use Dimensionality reduction (PCA)



Improvements

- Weighted voted:
  Give a weight to each neighbor. Weight inversely depends on distance or we can use kernels

- Define your own distance:
  Each feature has a distance
  Each feature distance has a weight

$$D(x_i, x_{i'}) = \sum_{j=1}^{p} w_j \cdot d_j(x_{ij}, x_{i'j}); \quad \sum_{j=1}^{p} w_j = 1.$$

From LogReg to KNN

LogReg:
- Low variance high bias
- Linear restriction

KNN:
- High variance and low bias
- No restriction
- Depends on training (selection of k)