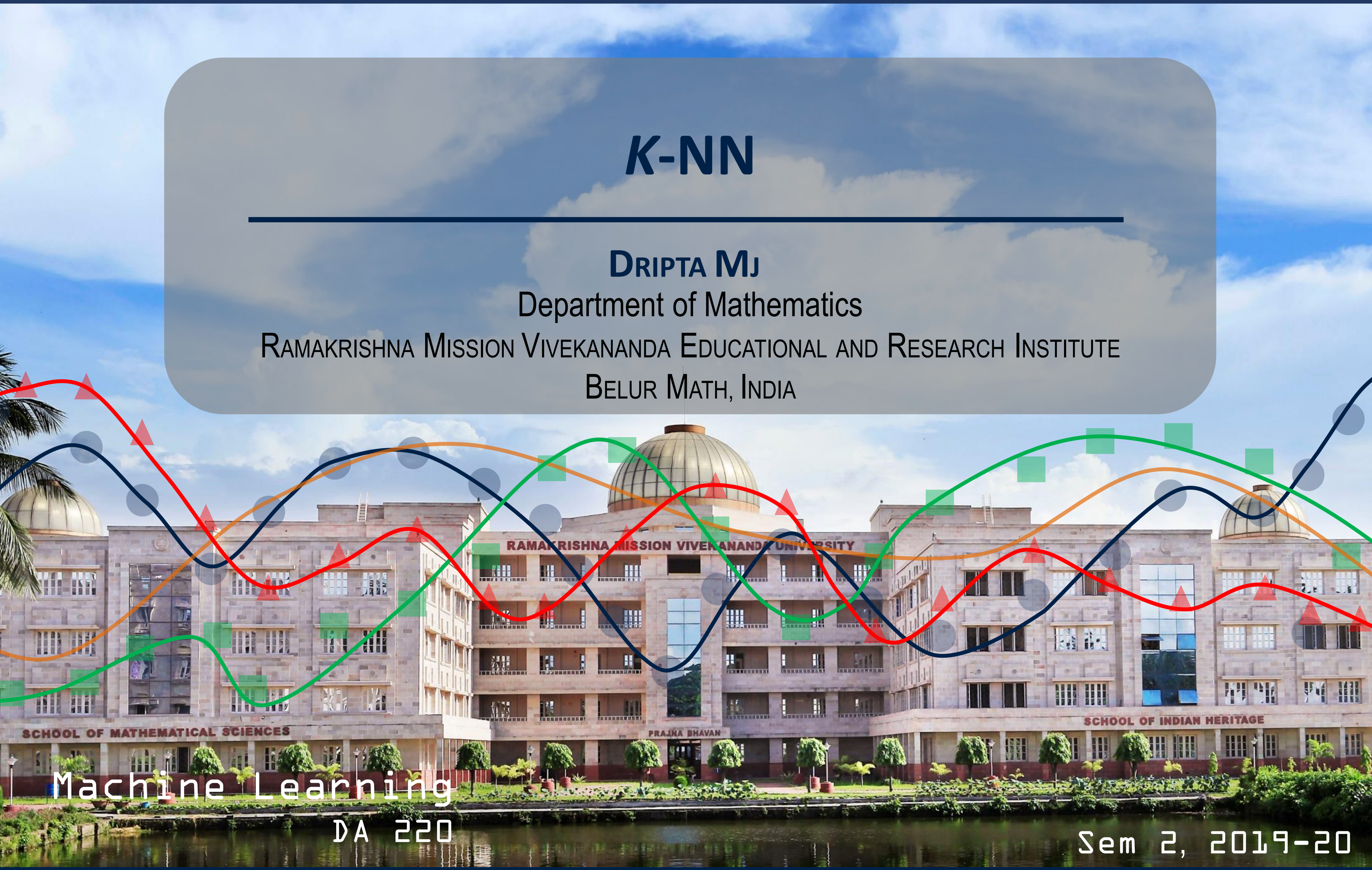
K-NN

Dripta Mj
Department of Mathematics
Ramakrishna Mission Vivekananda Educational and Research Institute
Belur Math, India



Machine Learning
DA 220

Sem 2, 2019-20

Introduction

- Supervised learning algorithm.

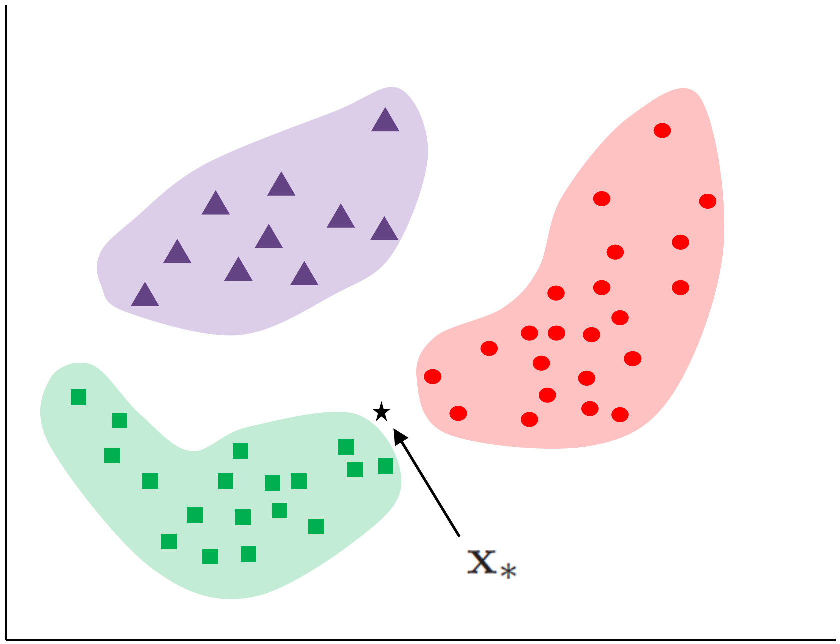
- Training dataset: $\mathcal{D} = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(1)}, y^{(1)}), ..., (\mathbf{x}^{(N)}, y^{(N)})\}$.

- Input data comprise D features. For example, the ith example

$$\mathbf{x}^{(i)} = \left(x_1^{(i)}, x_2^{(i)}, ..., x_D^{(i)}\right)$$

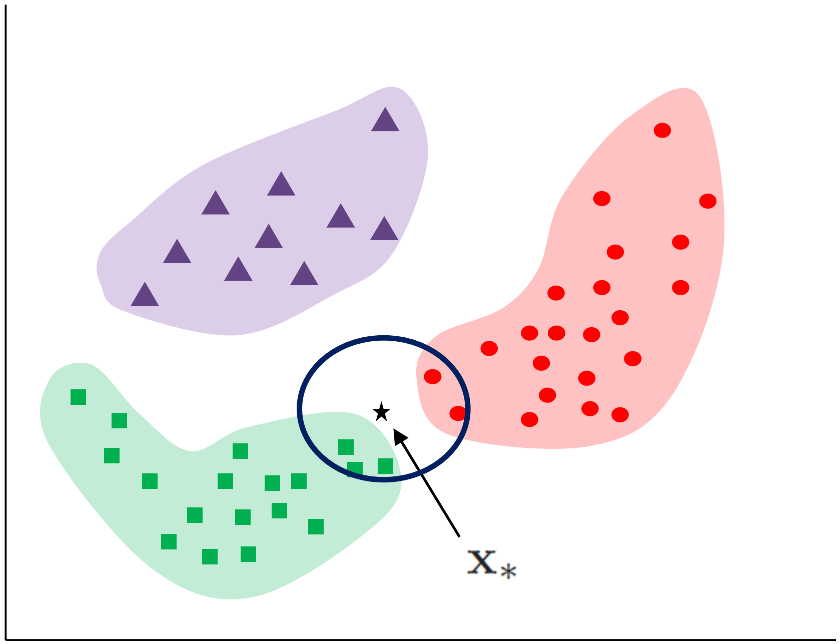
- Objective: Predict y_* for an unobserved example $\mathbf{x}_*$.

Example



- Problem: Assign class to $\mathbf{x}_*$.
- Prediction based on nearest K examples to $\mathbf{x}_*$.
  - Assign $\mathbf{x}_*$ to the class with the highest number of occurrences in the K nearest examples.

Example



- The algorithm needs a value of K.
- Suppose we take $K = 5$.
- Assign $\mathbf{x}_*$ to class ■ .

Mathematics

- Let $N_K(\mathcal{D}, \mathbf{x}_*)$ be the set comprising K closest points to $\mathbf{x}_*$ in $\mathcal{D}$.
- Prediction:

$$y_* = \arg\max_{c_j} \sum_{\mathbf{x}^{(i)} \in N_K(\mathcal{D}, \mathbf{x}_*)} \mathbb{1}_{(y^{(i)} = c_j)}$$

- $\mathbb{1}_z$ is the indicator function:

$$\mathbb{1}_z = \begin{cases} 1 & \text{if } z \text{ is true} \\ 0 & \text{if } z \text{ is false} \end{cases}$$

- Probabilistic modelling:

$$p(y_* = c_j | \mathcal{D}, \mathbf{x}_*, K) = \frac{1}{K} \sum_{\mathbf{x}^{(i)} \in N_K(\mathcal{D}, \mathbf{x}_*)} \mathbb{1}_{(y^{(i)} = c_j)}$$

- Assign $\mathbf{x}_*$ to the class with the highest probability.

Distance metric

- Euclidean distance

$$d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \sqrt{\sum_{p=1}^{D} (x_p^{(i)} - x_p^{(j)})^2}$$

$$= \sqrt{\sum_{p=1}^{D} \left(x_p^{(i)}\right)^2 + \sum_{p=1}^{D} \left(x_p^{(j)}\right)^2 - 2\sum_{p=1}^{D} x_p^{(i)} x_p^{(j)}}$$

$$= \sqrt{||\mathbf{x}^{(i)}||^2 + ||\mathbf{x}^{(j)}||^2 - 2\left(\mathbf{x}^{(i)}\right)^T \mathbf{x}^{(j)}}$$

- $||\mathbf{x}^{(i)}||$ is the norm of vector $\mathbf{x}^{(i)}$.
- $\left(\mathbf{x}^{(i)}\right)^T \mathbf{x}^{(j)}$ is the inner product of $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$.

Distance metric

- General distance metric – Minkowski distance

$$d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \left(\sum_{p=1}^{D} |x_p^{(i)} - x_p^{(j)}|^m \right)^{1/m}$$

  - $m = 2$ indicates Euclidean distance (l_2-norm)
  - $m = 1$ indicates Manhattan distance (l_1-norm)
  - $m \to \infty$ indicates Maximum Norm

- Hamming distance (categorical attributes):

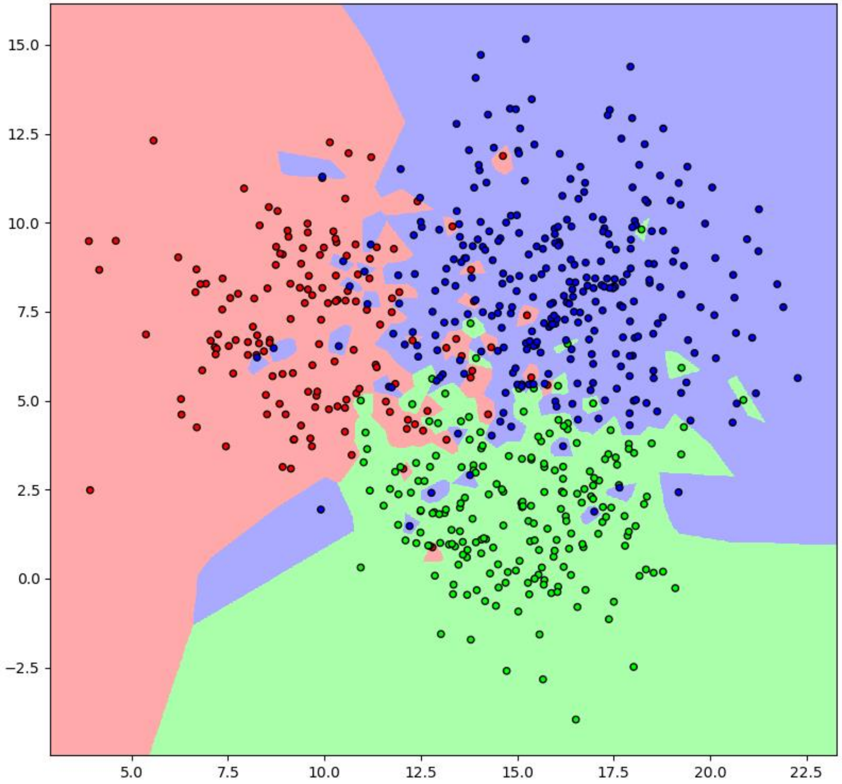
$$d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \sum_{d=1}^{D} \mathbb{1}_{\mathbf{x}_d^{(i)} \neq \mathbf{x}_d^{(j)}}$$

Procedure

- Find the class y_* of the new data point $\mathbf{x}_*$.
  - Compute the distance from $\mathbf{x}_*$ to all points in the training dataset.
  - Sort all the points based on their distance from $\mathbf{x}_*$.
  - Choose the K closest points to $\mathbf{x}_*$.
  - Find the class (label) with the most number of occurrence among the K nearest neighbours. Suppose that class is c_j.
  - Assign $\mathbf{x}_*$ to class c_j i.e. $y_* = c_j$.
- Note, features need to be normalized.
  - Standardize the inputs: Zero mean and unit variance.
  - For example, replace $x_p^{(i)}$ with $(x_p^{(i)} - \overline{x}_p)/\sigma_p$ where

$$\overline{x}_p = \frac{1}{N}\sum_{i=1}^{N} x_p^{(i)} \qquad \text{and} \qquad \sigma_p^2 = \frac{1}{N}\sum_{i=1}^{N}\left(x_p^{(i)} - \overline{x}_p\right)^2$$

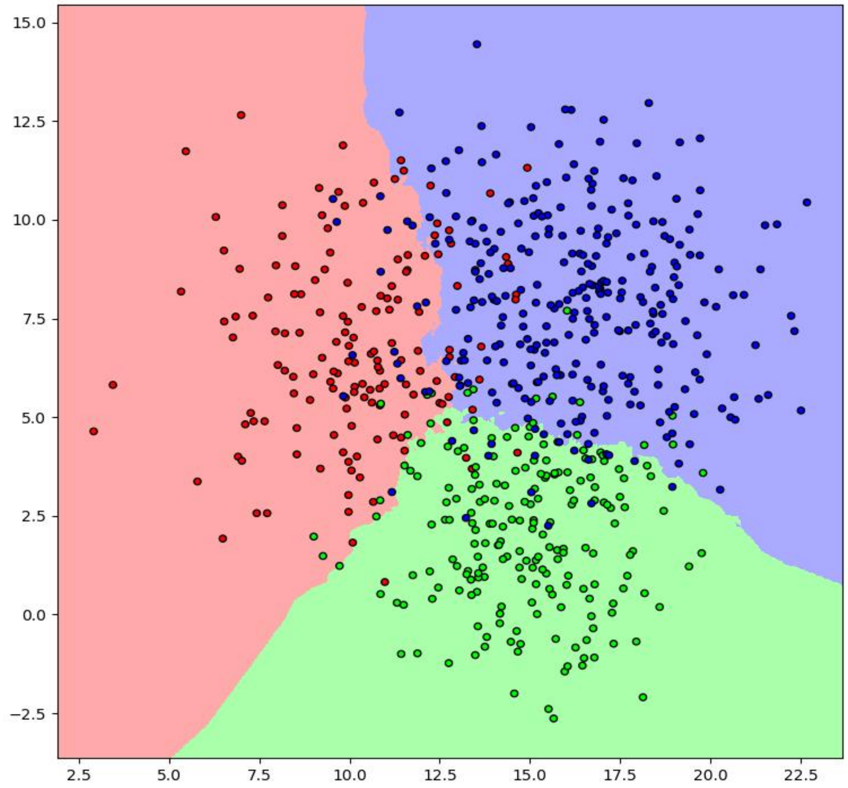
$K = 1$



- For small values of K:
  - Produces more number of small-sized regions for the classes.
  - Can lead to overfitting.

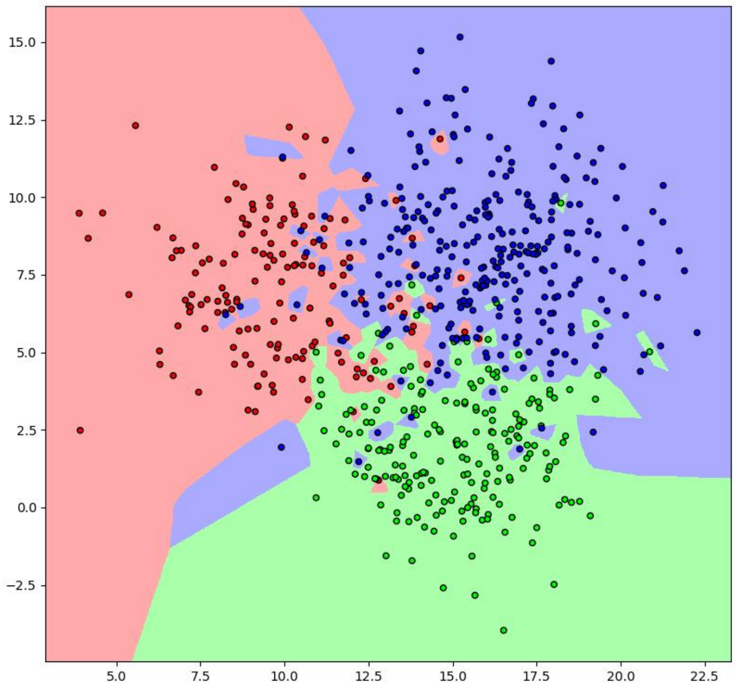
K value

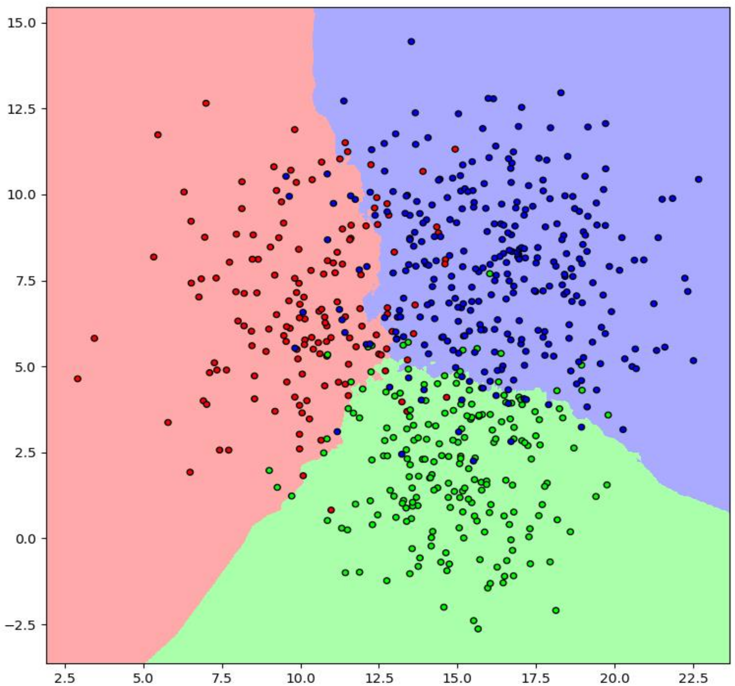
$K = 25$



- For large values of K:
  - Produces lesser number of regions.
  - Can lead to underfitting.

K value estimation

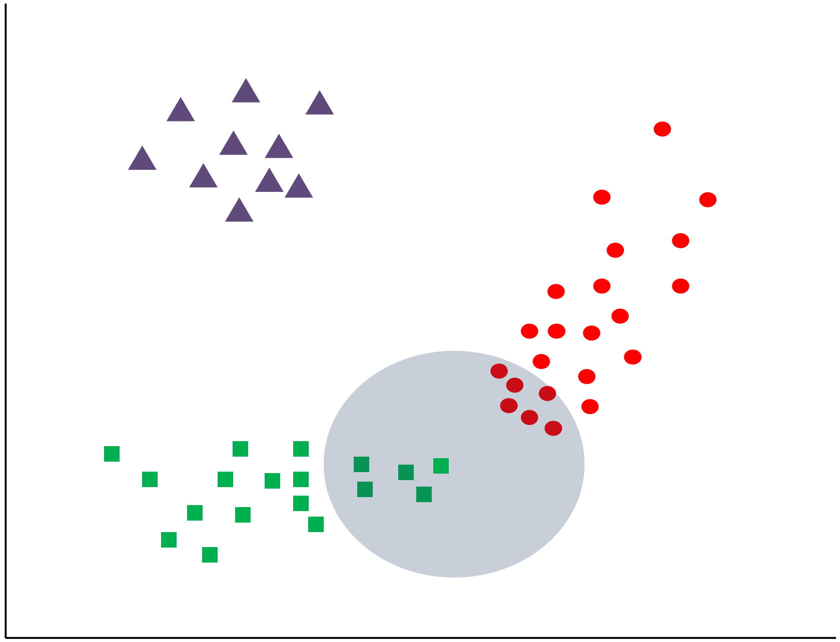




- Estimate K based on error on validation dataset or through cross-validation.

Weighted K-NN

- All the K nearest neighbours receive the same importance. However, points close to $\mathbf{x}_*$ should have more influence than those far away.



Procedure

- Let $N_K(\mathcal{D}, \mathbf{x}_*)$ be the set comprising K closest points to $\mathbf{x}_*$ in $\mathcal{D}$.

- Weighted K-NN: Each point in $N_K(\mathcal{D}, \mathbf{x}_*)$ is assigned a weight depending upon its distance from $\mathbf{x}_*$.
  - Let $w(\mathbf{x}, \mathbf{x}_*)$ be the weight assigned to $\mathbf{x} \in N_K(\mathcal{D}, \mathbf{x}_*)$.

- $w(\mathbf{x}, \mathbf{x}_*)$ is high if $\mathbf{x}$ is close to $\mathbf{x}_*$.

- $w(\mathbf{x}, \mathbf{x}_*)$ is low if $\mathbf{x}$ is far from $\mathbf{x}_*$.

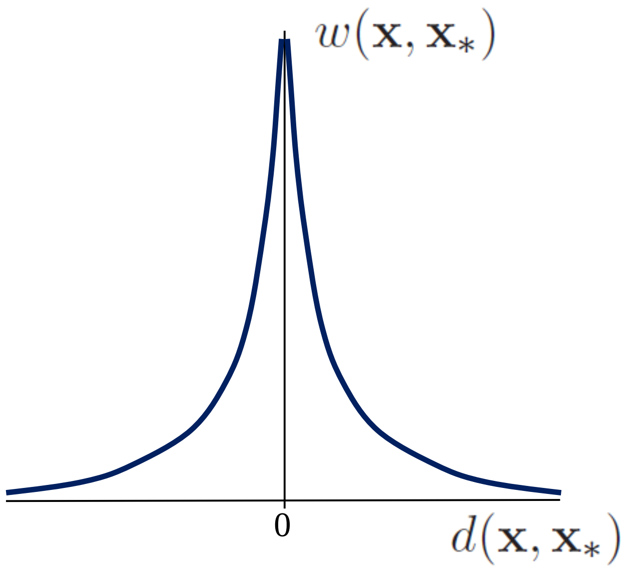
- Prediction:

$$y_* = \arg\max_{c_j} \sum_{\mathbf{x}^{(i)} \in N_K(\mathcal{D}, \mathbf{x}_*)} \mathbb{1}_{(y^{(i)} = c_j)} w(\mathbf{x}^{(i)}, \mathbf{x}_*)$$

Weight function

- Examples of weight functions:

$$w(\mathbf{x}, \mathbf{x}_*) = \frac{1}{d(\mathbf{x}, \mathbf{x}_*)}$$



$$w(\mathbf{x}, \mathbf{x}_*) = \exp\left(-\frac{d(\mathbf{x}, \mathbf{x}_*)^2}{\sigma^2}\right)$$