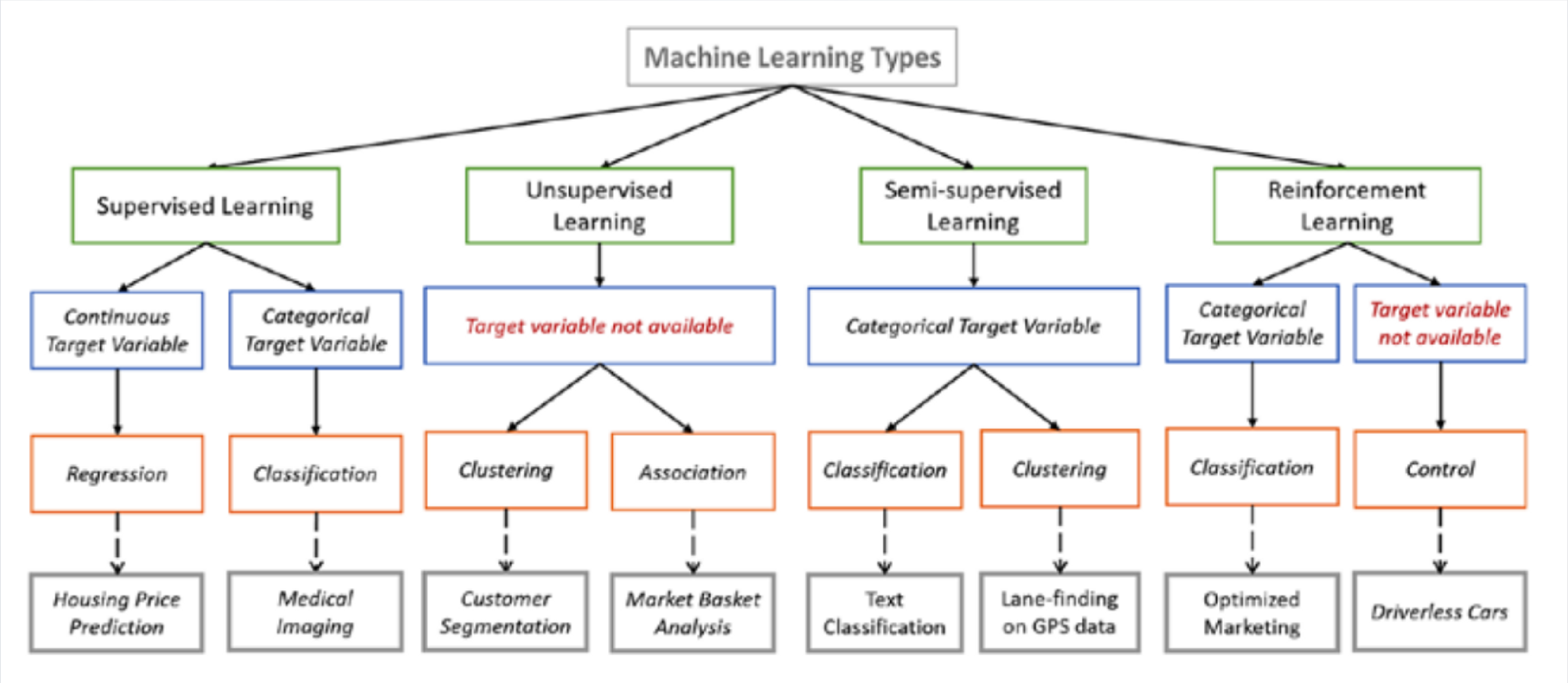
ML diagram



(from Sengupta et al. 2020)

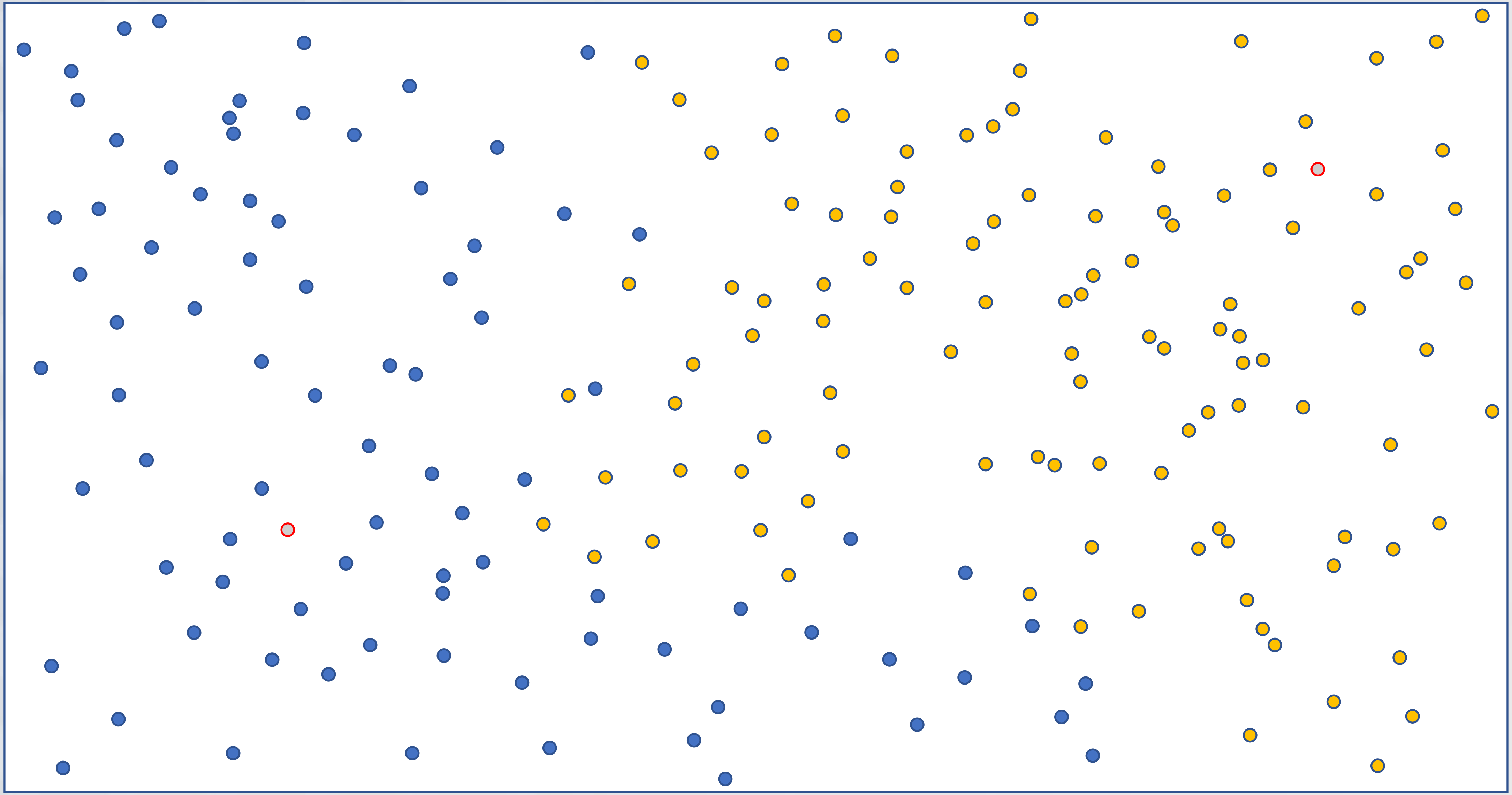
Week 6

- Regression revisited
  - Polynomial regression
  - Model selection
  - Overfitting
  - Periodic functions
  - a bit about logistic regression
- K-Nearest Neighbours
- Support Vector Machines
- Decision trees

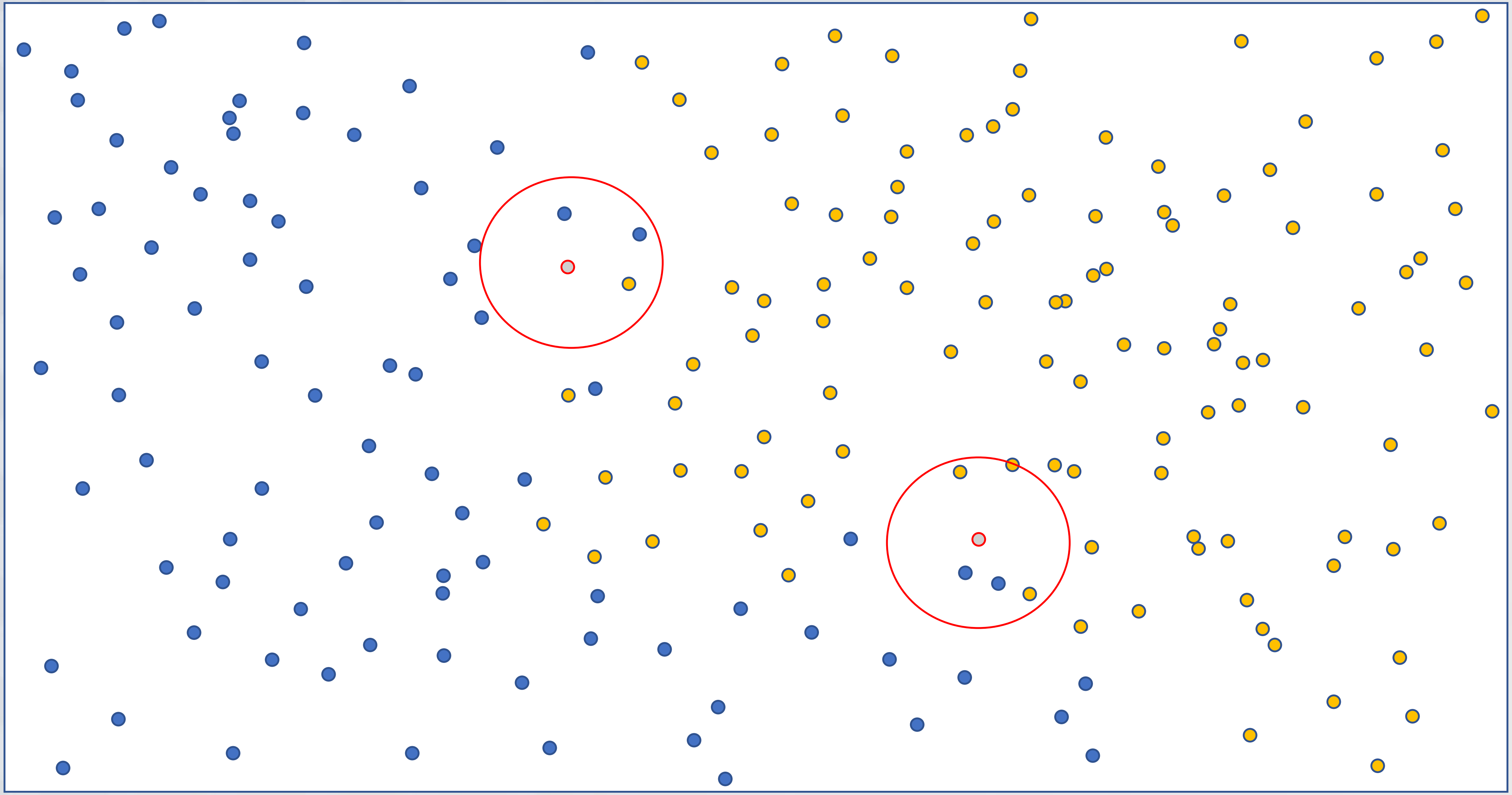
Continuous target variable

Discrete target variable

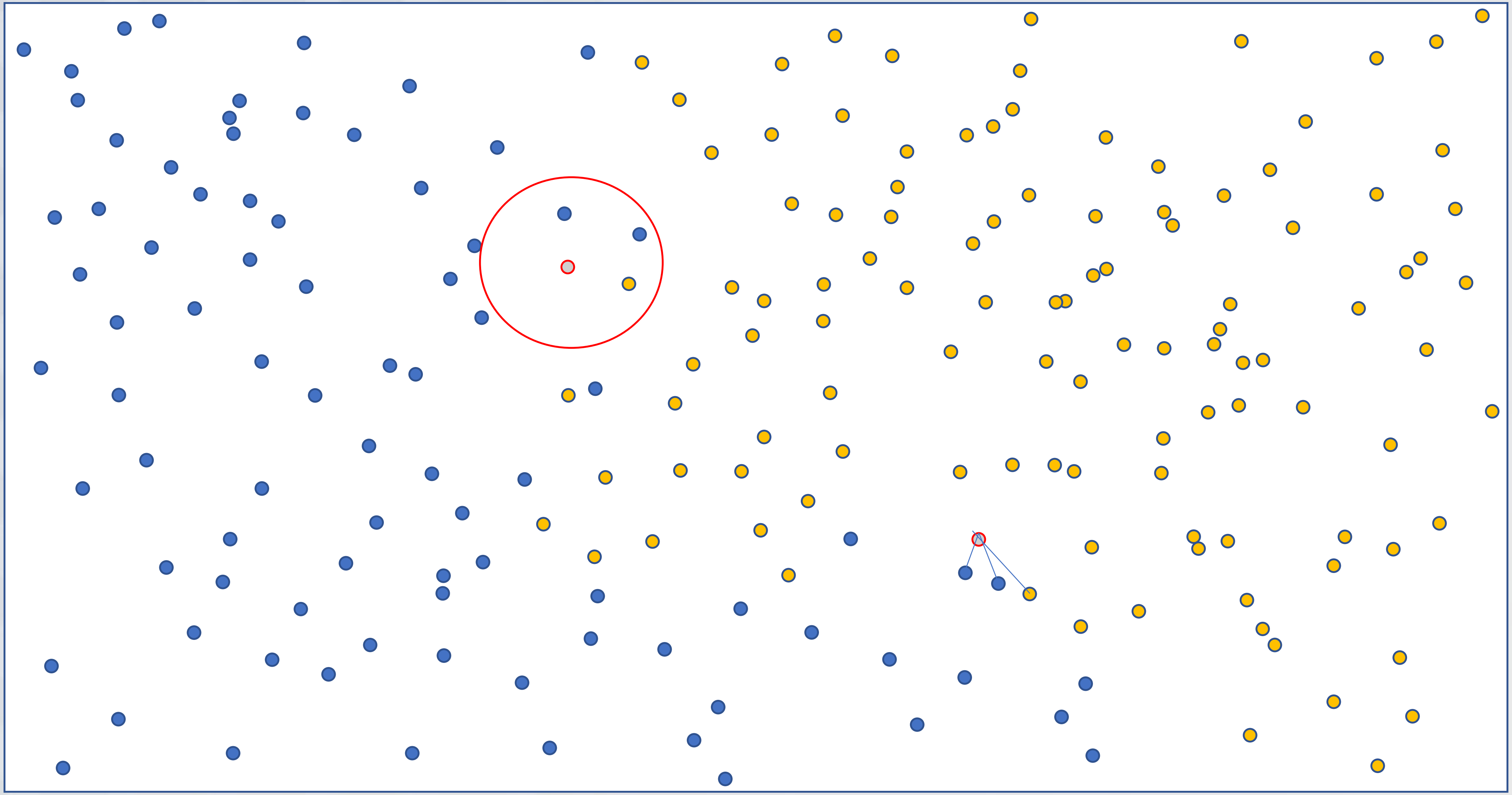
Example 1



Example 2



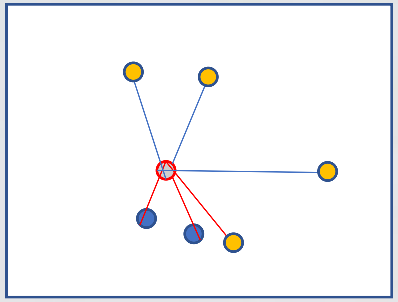
Example 2



K Nearest Neighbours (kNN) algorithm

- *Labelling by association*

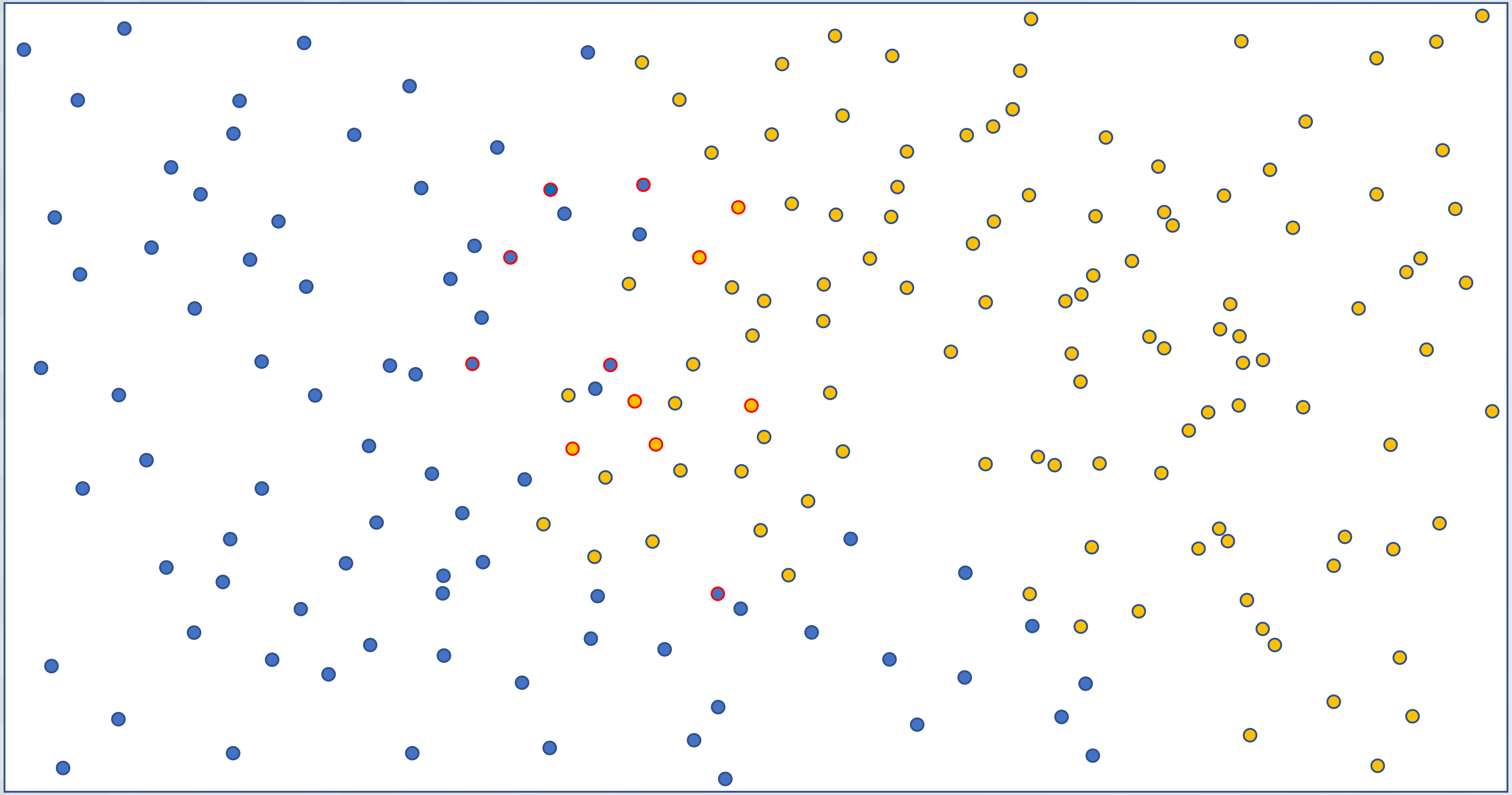
- We have a training data set consisting of N entries $\boldsymbol{X}_i, P_i$, where $\boldsymbol{X}_i$ is the position and P_i is the label. We need to evaluate P_i – the label for some test data entry with known position $\boldsymbol{X}_t$. P_t of that test data entry will be the same as the prevalent label of its k nearest neighbours.



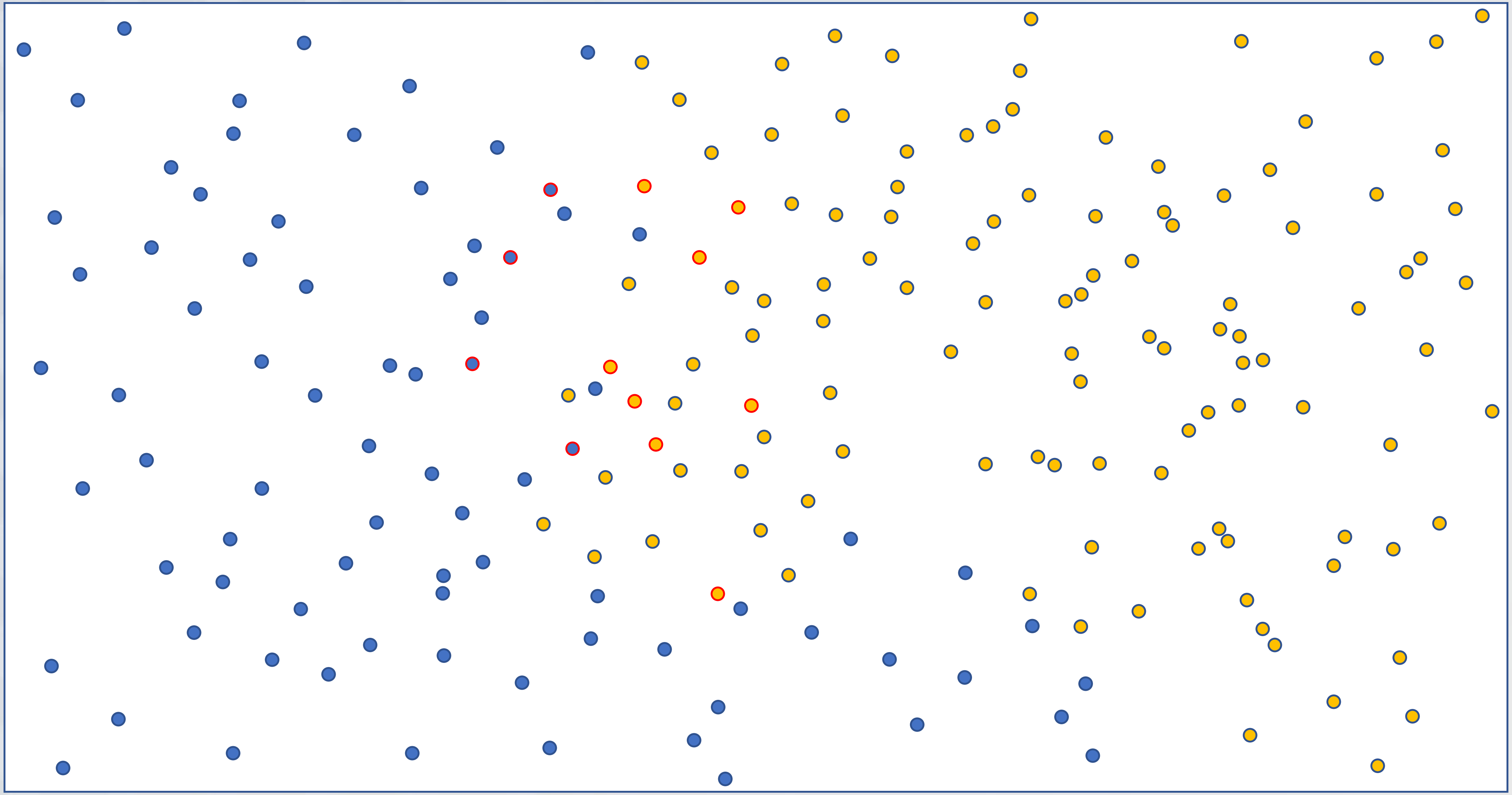
Basic algorithm (without optimisation)

- Assume P can be -1 or 1
- Measure N distances $L_i = \sqrt{(X_t - X_i)^2}$
- Sort linked arrays L and P so that L is in ascending order (values increasing)
- $P_t = \mathrm{sign}(\sum_0^{k-1} P_i)$
- For test data consisting of M entries and training data consisting of N entries we will need to evaluate M×N distances. Obviously, for large data sets we need some optimisation

Lower k



Higher k



Choice of k

- It affects the fine structure of your distribution
- Low k = fine structure retained but with higher noise level
- High k = Smooth (low noise level) but fine structure is lost
- Optimal choice of k depends on what you know about your data
- One possible strategy:
  - you what to keep the structure with the characteristic length of l;
  - you evaluate the average number of points (entries) K within the area πl^2,
  - use K as a parameter for kNN

Weighting

- The effect of neighbours may change with distance. We might want neighbours which are further away to have lower significance. How do we do this? – Weighting!
- $P_t = \text{sign}(\sum_0^{k-1} g_i P_i)$
- In our basic example $P_t = \text{sign}(\sum_0^{k-1} P_i)$ weights g_i=1
- Weight can decrease with the distance, e.g.
  $P_t = \text{sign}(\sum_0^{k-1} e^{-L_i} P_i)$
  in this case weights exponentially decrease with distance to the corresponding neighbour

‘Curse of dimensionality’

- The method becomes inefficient when dimensionality of the problem is high.

- Imagine a case when the number of dimensions is not much lower than the number of entries in the test data -> too many neighbours at similar distances