Supervised methods
k-Nearest Neighbors

Dr. Jakub Kuzilek

Introduction

Small recap: What is Supervised learning?

Introduction

Small recap: What is Supervised learning?

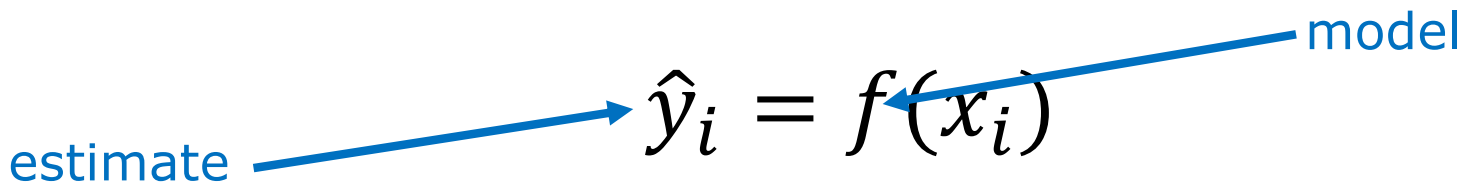
- We know the right answers
- Supervised learning algorithm is inferring decision function from **labelled** training data

Supervised learning

- We know the right answers
- Supervised learning algorithm is inferring decision function from **labelled** training data
- Input:

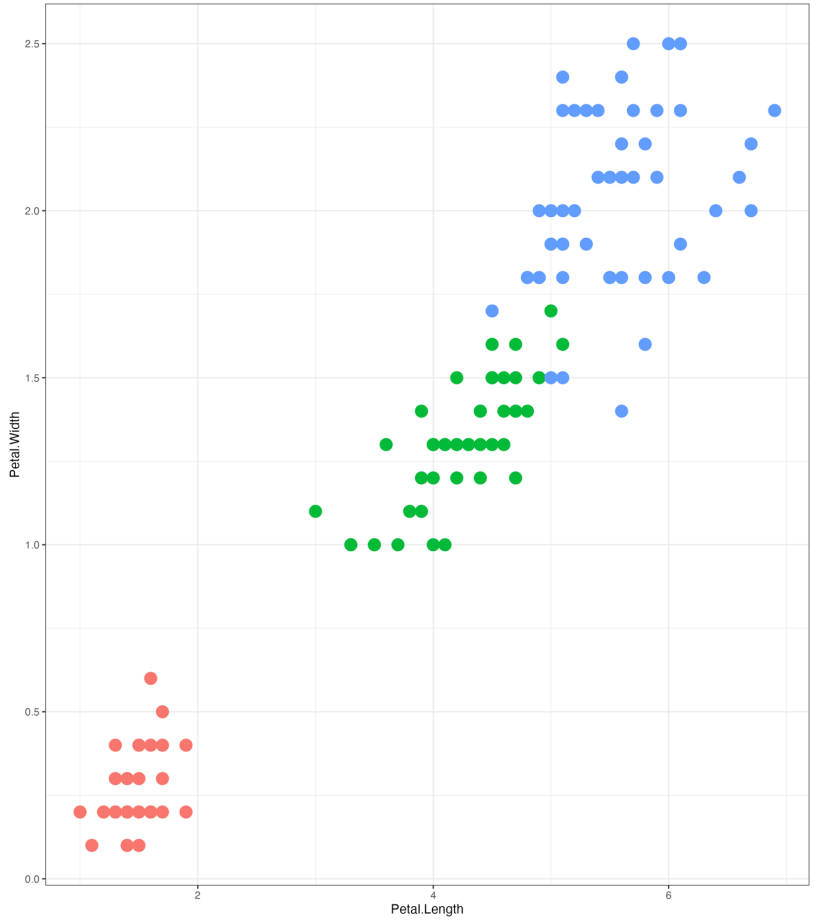
$$T = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots, (x_N, y_N)\}$$

- Output:



Classification

- Model created by selected algorithm assign the data specific categories
- Common methods:
  - Linear classifiers
  - Support Vectors Machines
  - Decision trees
  - Ensemble algorithm
  - k-Nearest Neighbours
  - Neural Networks
  - …



Distance function

Small recap: What is distance function and what properties it has?

Distance function

Small recap: What is distance function and what properties it has?

1. $d(x, x) = 0$
2. $d(x, y) \geq 0$
3. $d(x, y) = d(y, x)$
4. $d(x, z) \leq d(x, y) + d(y, z)$

Algorithm

- Training set:
$$T = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_N, y_N)\}$$

- Distance function $d: X \times X \rightarrow \mathbb{R}_o^+$

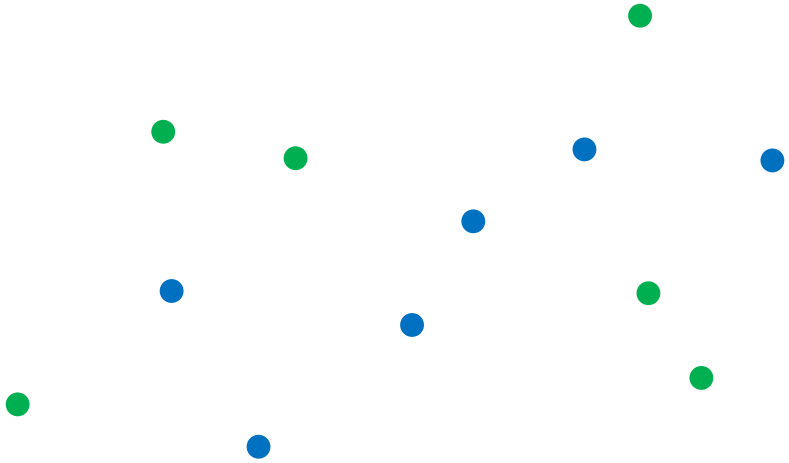
- Overall objective: find k samples $S = \{(x_{r1}, y_{r1}), \dots (x_{rk}, y_{rk})\}$ from T, which are closest to input sample x in sense of metric d

Algorithm

- To classify new sample x:
  1. Compute $d(x, x_i)$, where $x_i \in T$
  2. Make rank r_i of each sample x_i based on $d(x, x_i)$
  3. Take first k samples $S = \{(x_{r1}, y_{r1}), \dots (x_{rk}, y_{rk})\}$ from the ordered set of samples by the rank r_i
  4. Classify x to the class y, in which the most samples in S belong to.

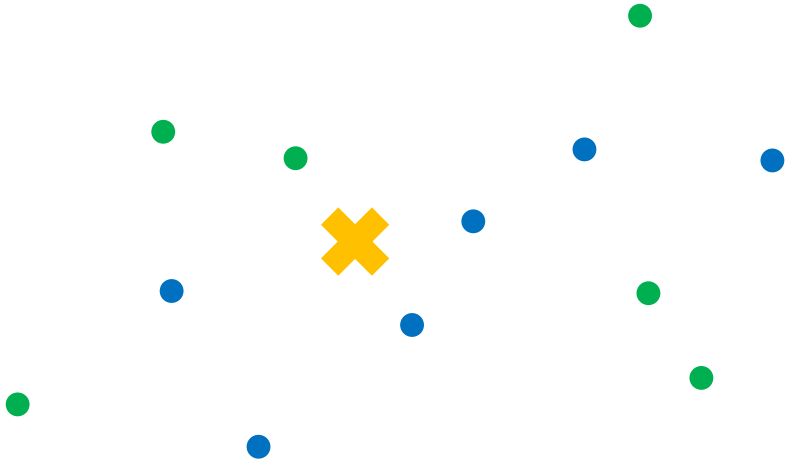
Algorithm

3-NN



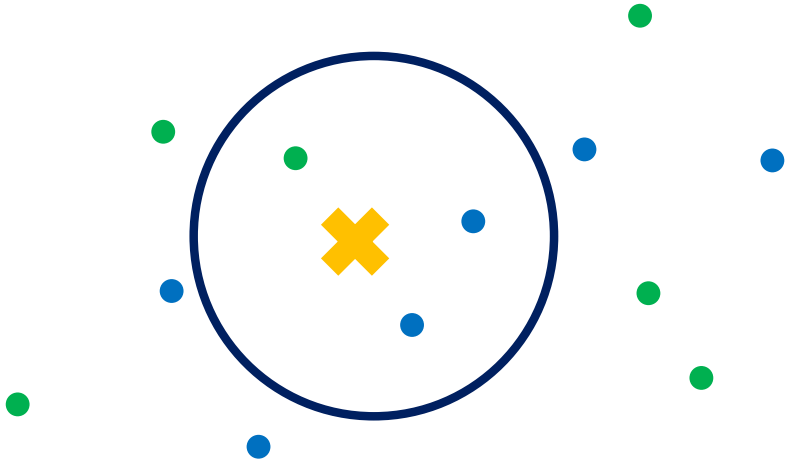
Algorithm

3-NN



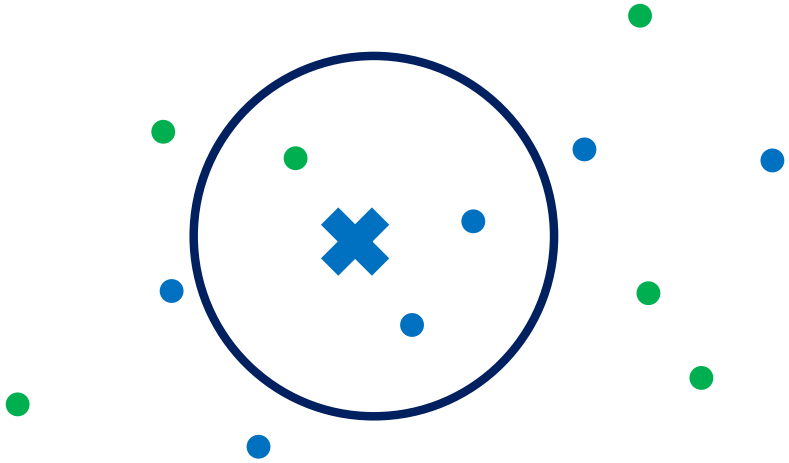
Algorithm

3-NN



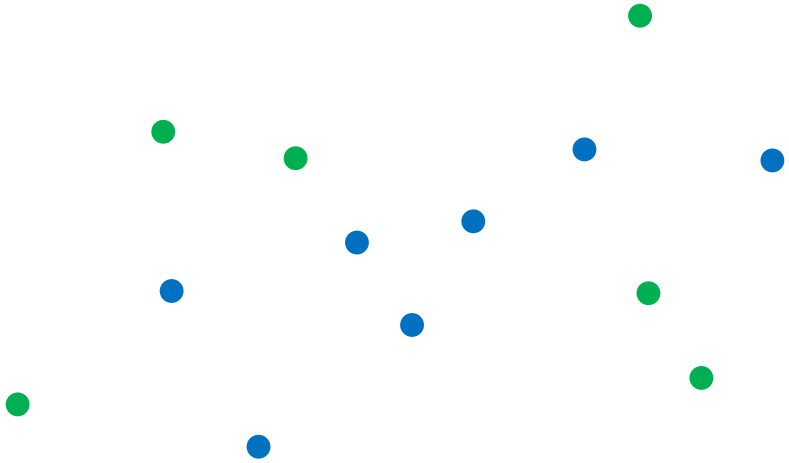
Algorithm

3-NN



Algorithm

3-NN



Properties

- trivial implementation
- error is bounded:
  - 1-NN

Naïve Bayes classifier error

$$\epsilon_B \leq \epsilon_{NN} \leq 2\epsilon_B - \frac{R}{R-1}\epsilon_B^2$$

number of classes

  - k-NN

$$\epsilon_{kNN} \leq \epsilon_B + \frac{\epsilon_{NN}}{\sqrt{k * const}}$$

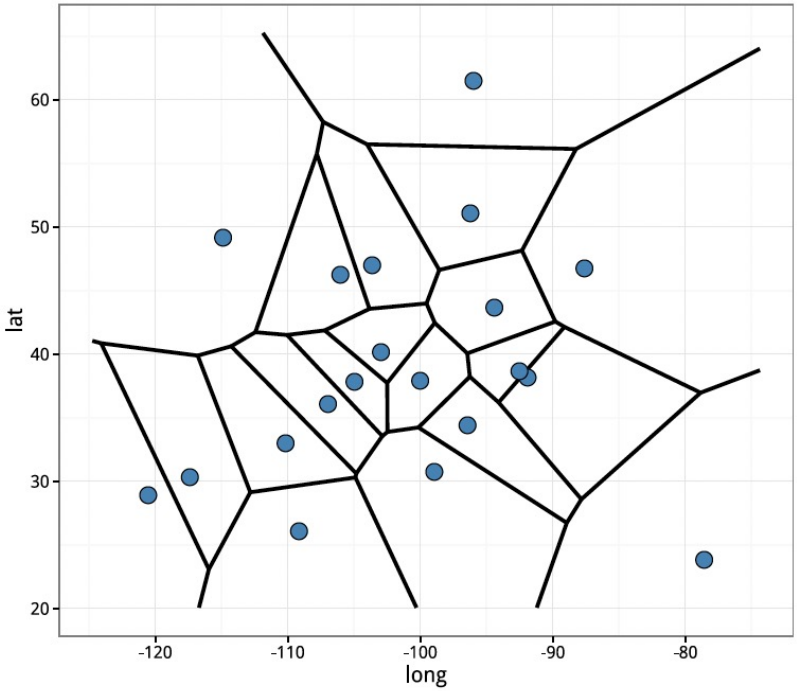
- slow when using naïve implementation
- High memory requirements
- Depends on metric d
- No generalization ability (tends to overfit, high variance)

Speedup

- Remove ”unimportant” samples from T, which does not influence the classification result
- Better algorithm for nearest neighbours' search

Voronoi diagram

- Set of regions dividing plane
- each region corresponds to one of the generators (points)
- All points in region are closer to generator of that region then to the any other set
- Can be represented as tree (Kirkpatrick’s point location algorithm)
- Nearest neighbours are searched by searching corresponding tree



Condensation

- reduction of T

- Algorithm:
  1. Create two lists A and B
  2. Select randomly sample from T and put it in A, all other samples go to B
  3. Classify B using 1-NN algorithm
  4. All misclassified samples from B goes to A
  5. Repeat (go to 2.) until all samples in B are correctly classified

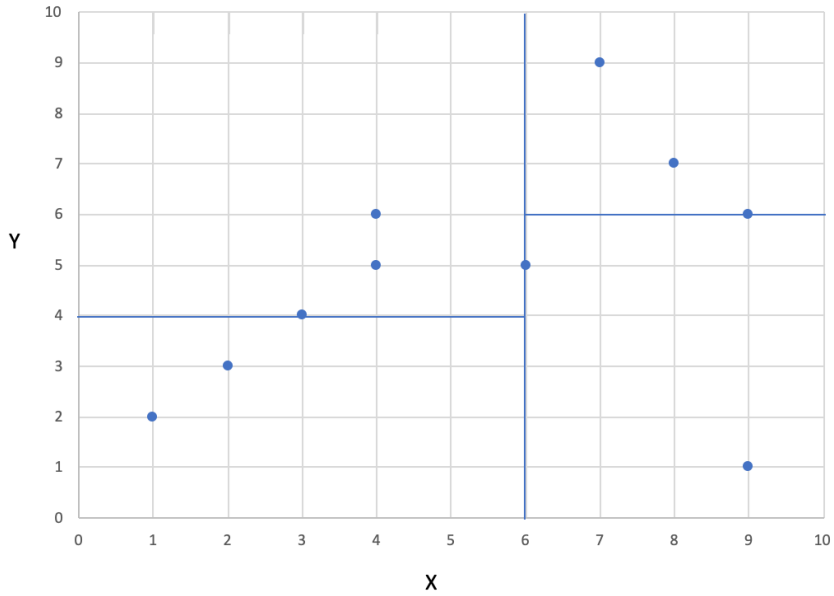
List A contains the condensed training set.

k-D trees

- Construct binary tree
- Each node is k-dimensional point
- Uses median value of points in one dimension

- Algorithm:
  1. Find median in one dimension
  2. Split the dataset based on that
  3. For “halfplanes” find medians in the following dimension
  4. Continue from 2.

- Search is then done via tree



https://towardsdatascience.com/tree-algorithms-explained-ball-tree-algorithm-vs-kd-tree-vs-brute-force-9746debcd940

Example

- Training set (test1, test2; course_outcome)
  $T = \{(7{,}7; passed), (7{,}4; passed), (3{,}4; failed)(1{,}4; failed)\}$
- Number of clusters
  $k = 3$
- Euclidean distance
- New sample $x = (3{,}7)$

Example

Sample (i)	Class	x_i	$d(x, x_i)$
1	*passed*	$(7; 7)$	$(7-3)^2+(7-7)^2= 16$
2	*passed*	$(7; 4)$	25
3	*failed*	$(3; 4)$	9
4	*failed*	$(1; 4)$	13

Example

Sample (i)	Class	$d(x, x_i)$	Rank (r_i)
1	*passed*	16	3
2	*passed*	25	4
3	*failed*	9	1
4	*failed*	13	2

Example

Sample (i)	Class	$d(x, x_i)$	Rank (r_i)
1	*passed*	16	3
2	*passed*	25	4
3	*failed*	9	1
4	*failed*	13	2

Example

Sample (i)	Class	$d(x, x_i)$	Rank (r_i)
1	*passed*	16	3
2	*passed*	25	4
3	*failed*	9	1
4	*failed*	13	2

- Final class y is majority from:

Class	Count
passed	1
failed	2

$\boldsymbol{failed}$

Questions?