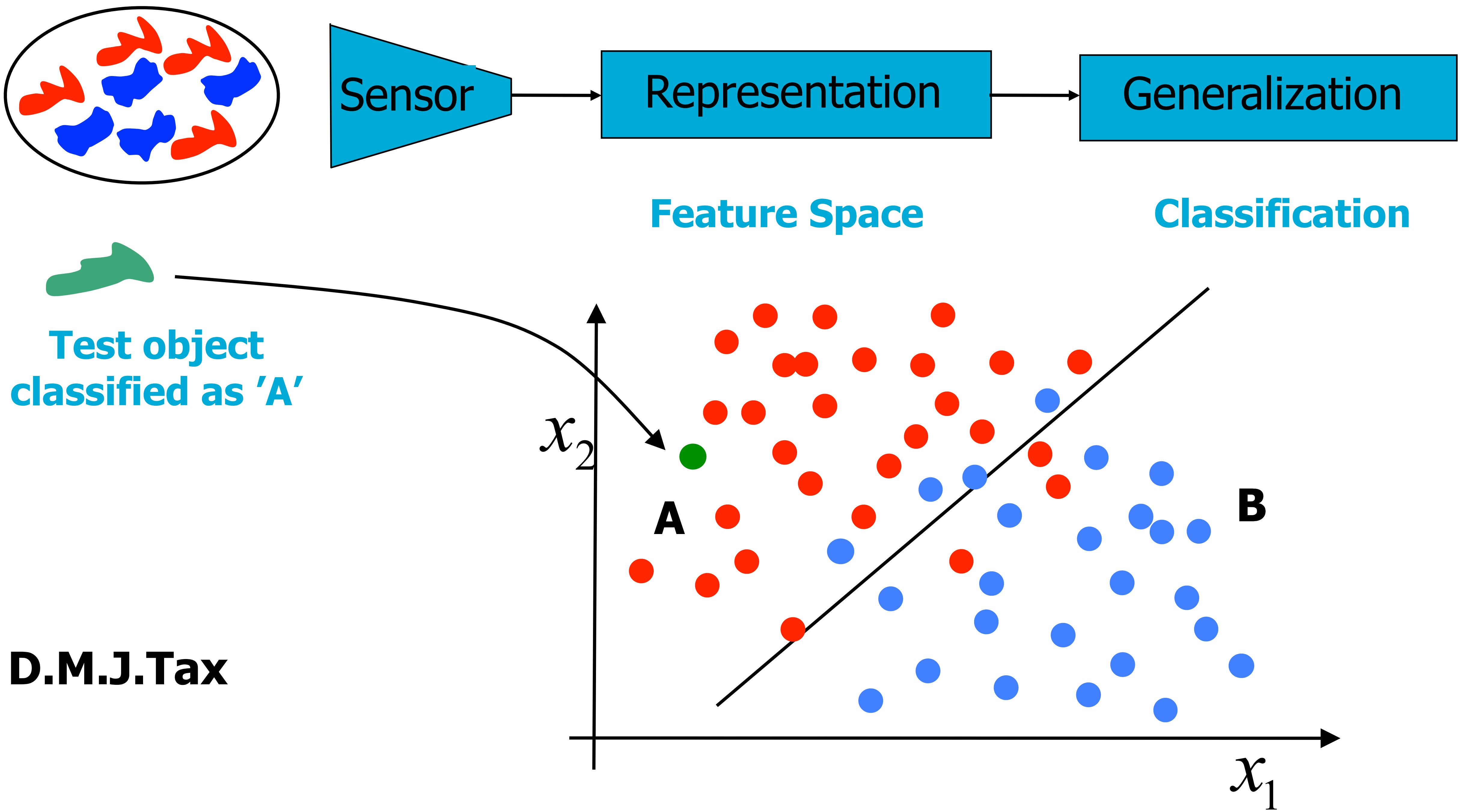
Machine Learning Experiments



D.M.J.Tax

TU Delft

Contents

- Hour 1: Some small things (use this for your project as well!):
  - Practical data analysis
  - Stratified cross-validation, leave-one-group-out
  - Paired t-test
  - Hyperparameter optimisation
  - Presentation of results
- Hour 2: Fairness in ML
  - Intuition
  - Formalization of fairness criteria
  - Issues

The final project

- Submit the report and code to Brightspace
- The deadline is mentioned in the Assignments (January 23, 2025)

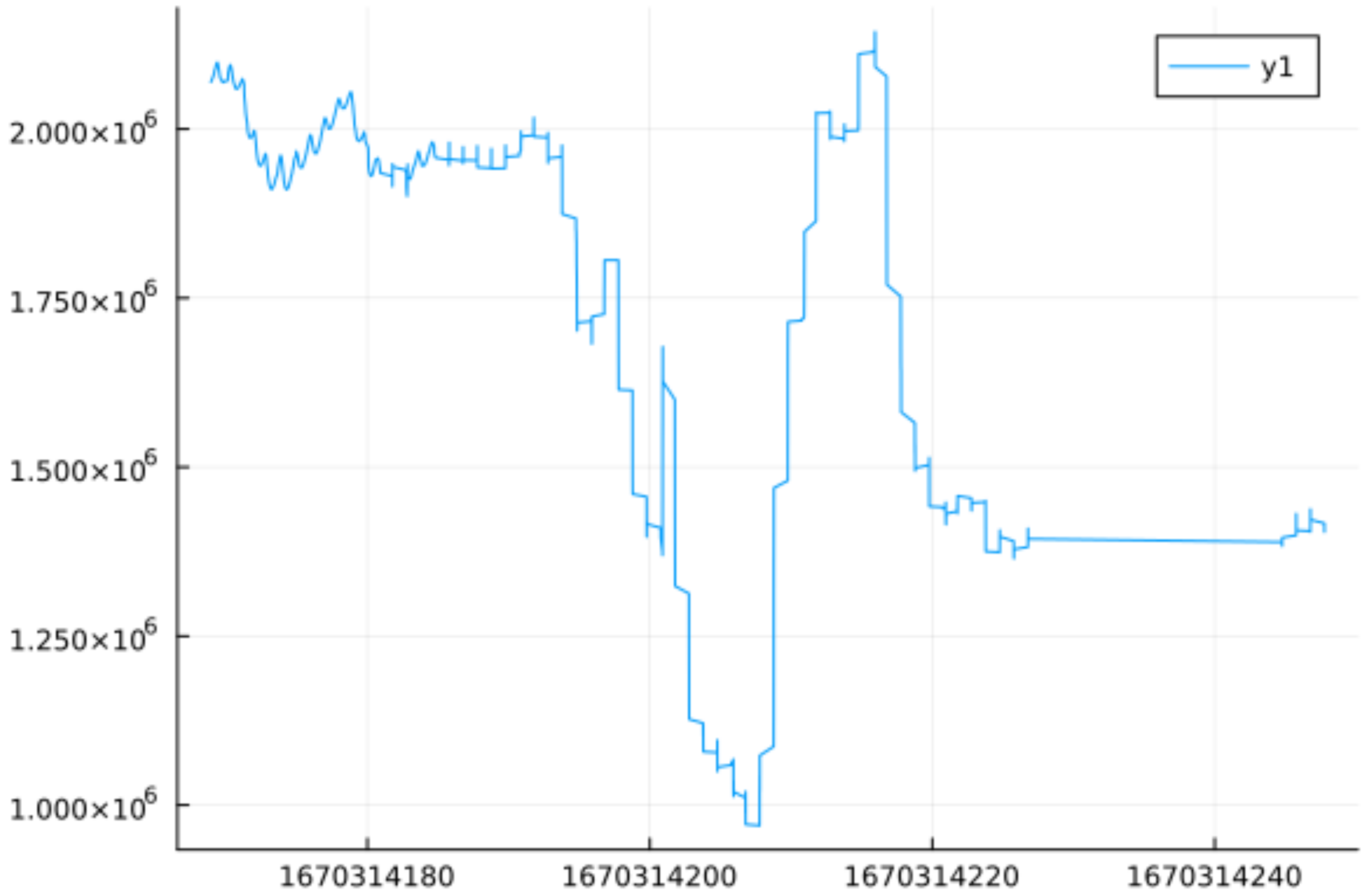
- Try to make it a Machine Learning problem, not a data analysis problem
- Have a look at the chapter 'Final Project' in the exercise manual
- Focus on the comparison between ML methods, their strengths and weaknesses, and not so much on solving one particular classification/regression problem

Trial Exam

- On weblab.tudelft.nl a practice exam is available
- Try it!
- In the lecture in January we'll have a Q&A session, we can also these trial exam

Inspect your data

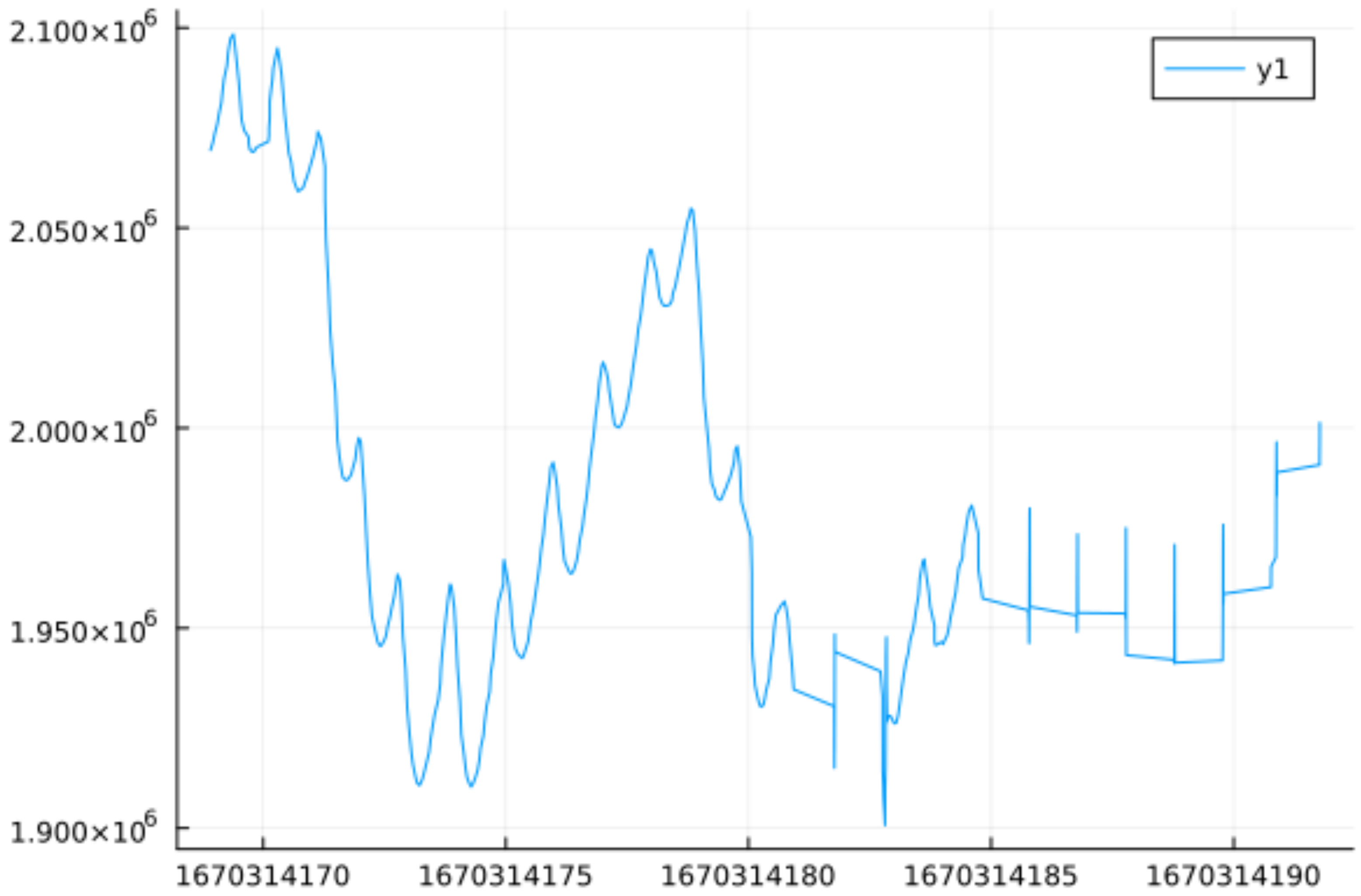
- Data from Fitbit wearable; a few minutes of PPG signal:



TU Delft

Zoom in of Fitbit data

- First part looks ok-ish, but after 10 seconds…

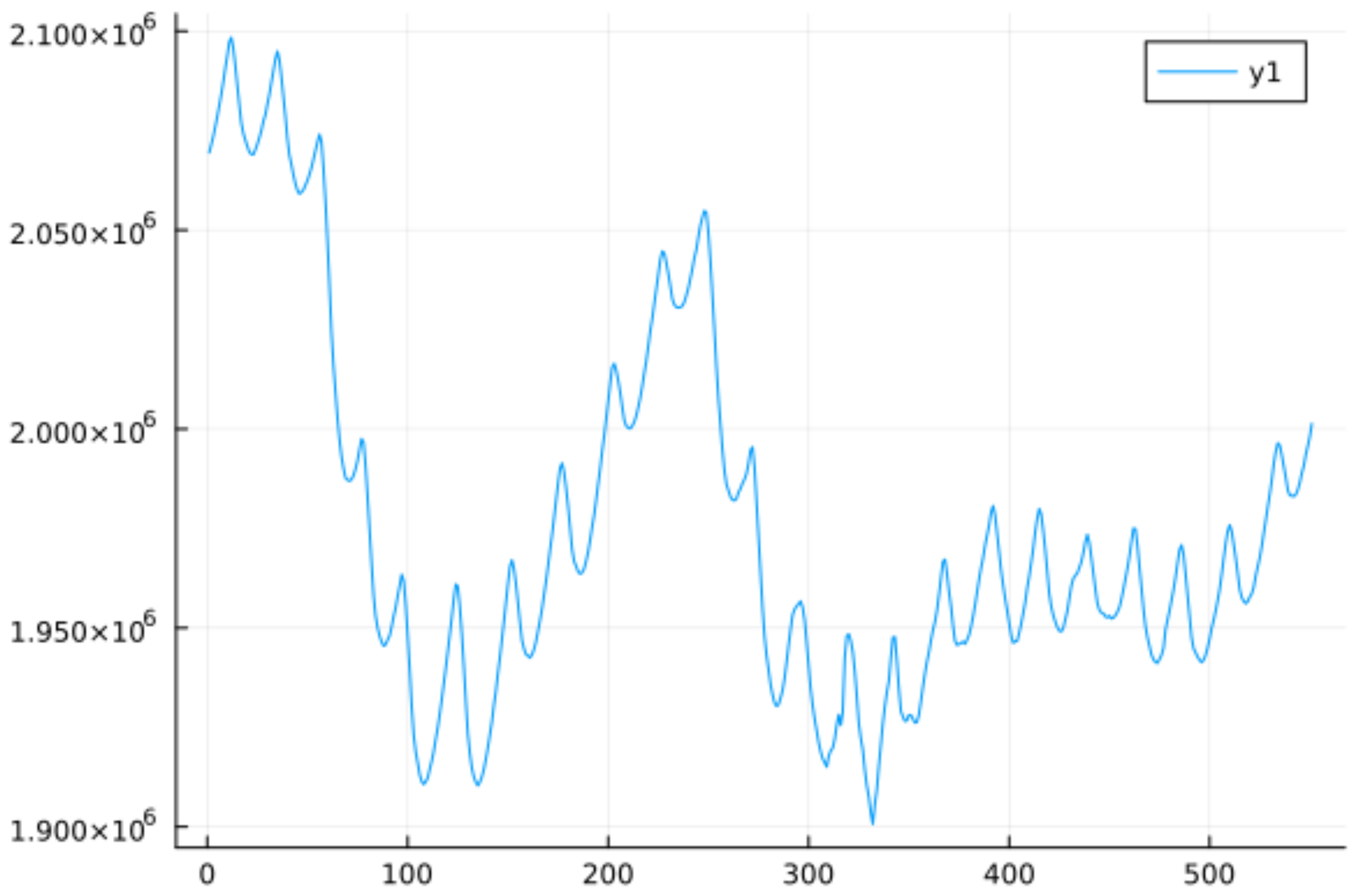


Practical data analysis

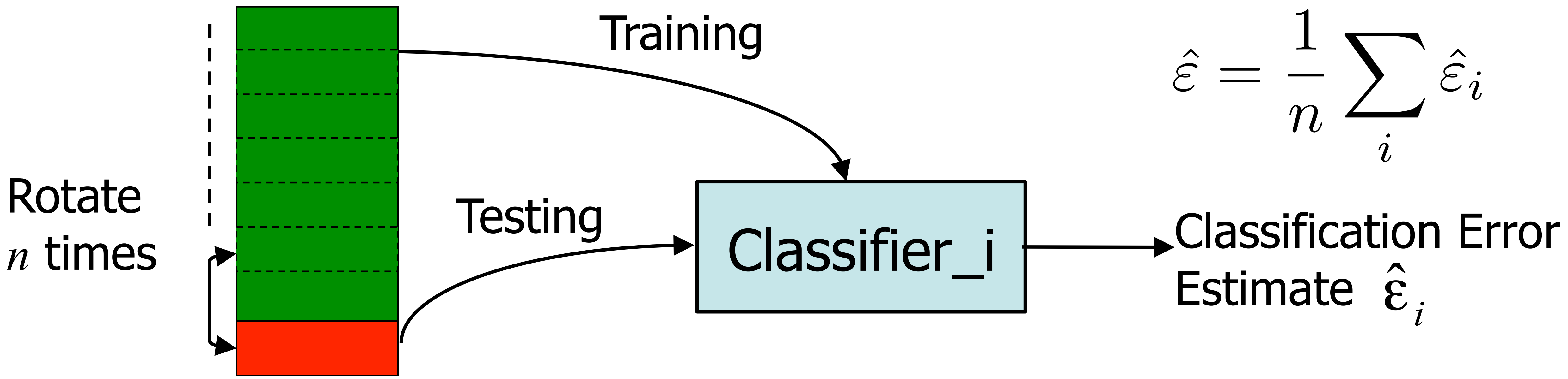
- Define what you want to predict, the target
- Split your data in a training/design set, and a test set (for smaller datasets, use crossvalidation)
- Eyeball your data (outliers, weird stuff), visualise with scatterplots
- Scale features, invent new features
- Try simple models (linear, nearest neighbour), if needed, extend to more complex
- Be paranoid, use sanity checks
- Which objects are wrongly predicted? Which classes?
- Do a sensitivity analysis ("ablation study")

Zoom in of Fitbit data

- Ignore the time-stamp: (bug in the hardware??)



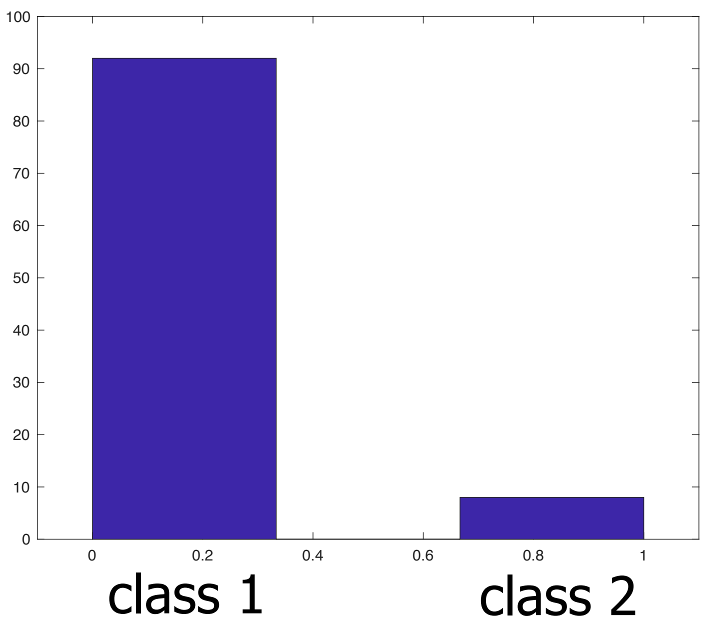
Cross-Validation



- Problems when classes are heavily imbalanced
- Problems when samples are very correlated

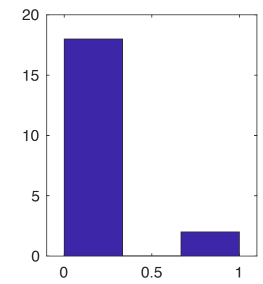
Standard cross-validation

full dataset:

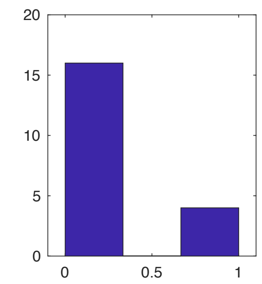


naive:

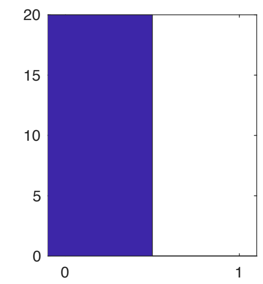
fold 1:



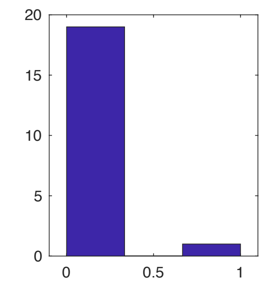
fold 2:



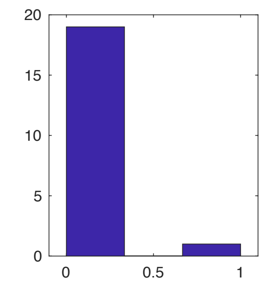
fold 3:



fold 4:

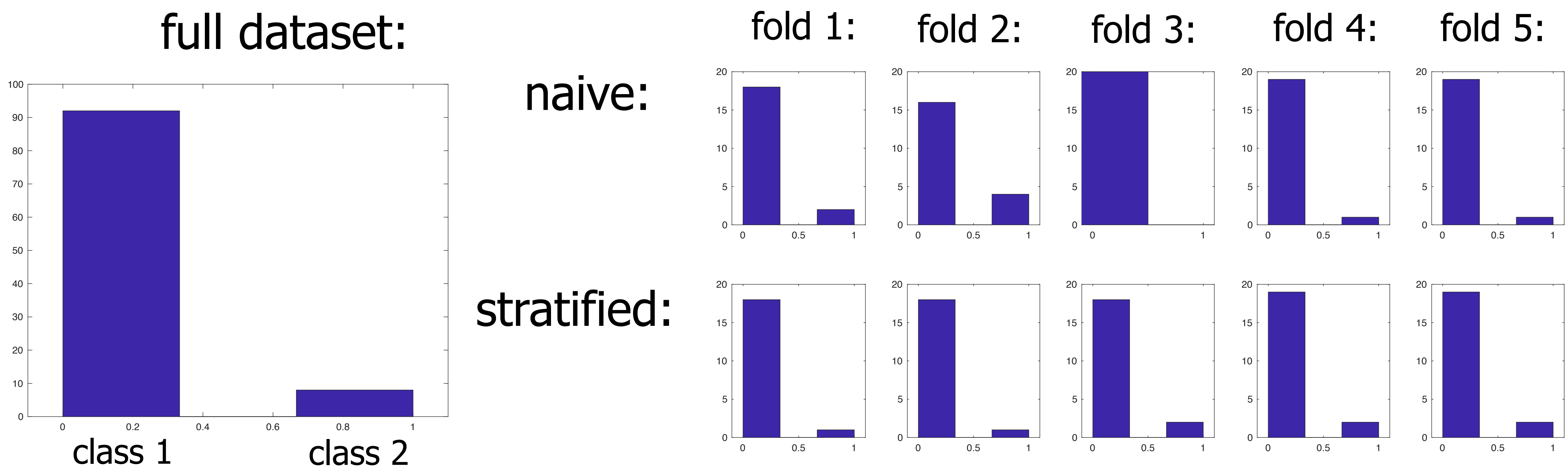


fold 5:



- Imbalanced classes: run the risk that the small class is lost in some of the folds

Stratified cross-validation

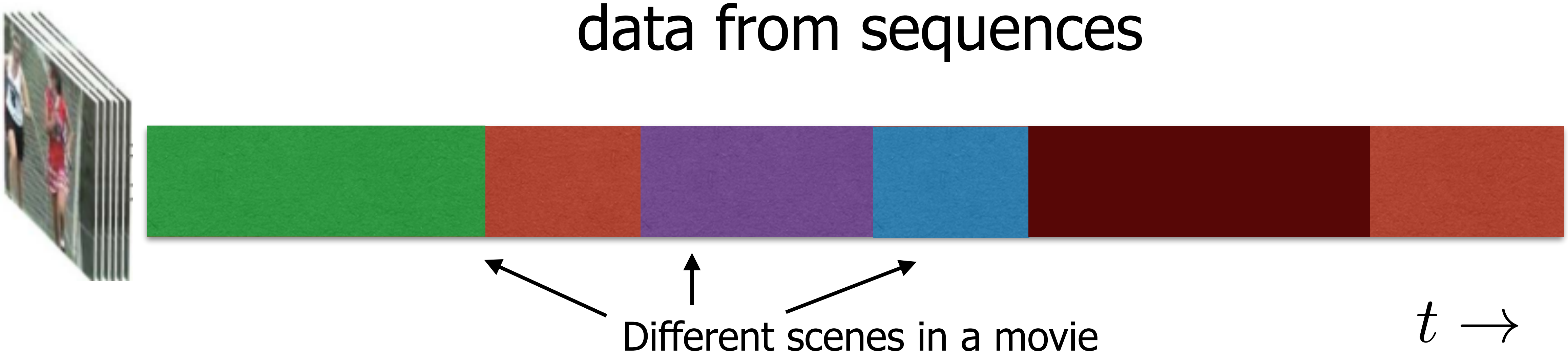


- Imbalanced classes: run the risk that the small class is lost in some of the folds
- Try to get the same distribution in each fold:
- 1. get the data from one class
- 2. split in K folds
- 3. combine the data from the different classes

TU Delft

Leave-one-group-out cross-validation

- For instance: data from different persons
  data from different videos
  data from sequences

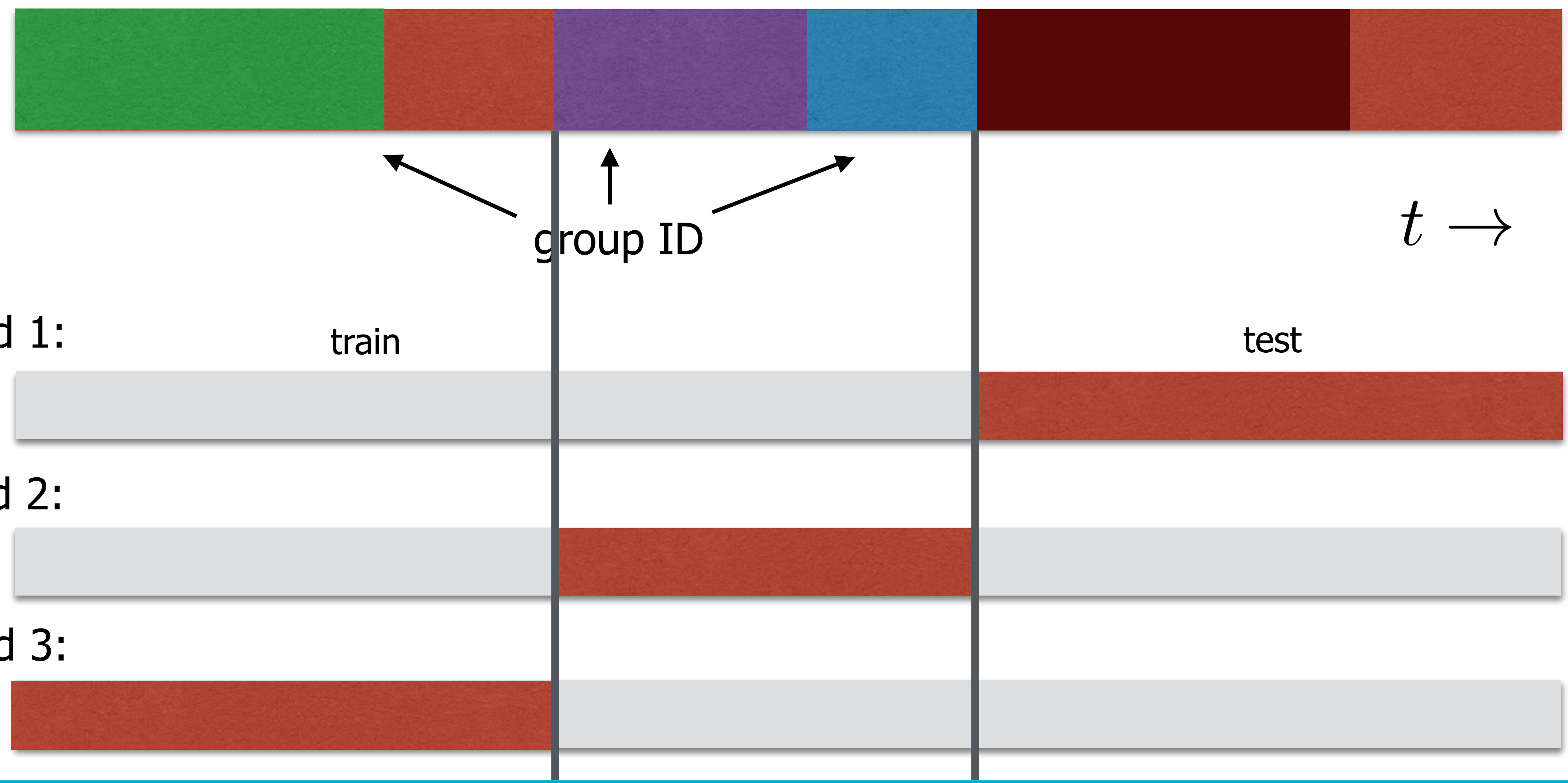


- Samples within a group are heavily correlated
- Mixing them in training and testing gives (too) optimistic performance estimates

TU Delft

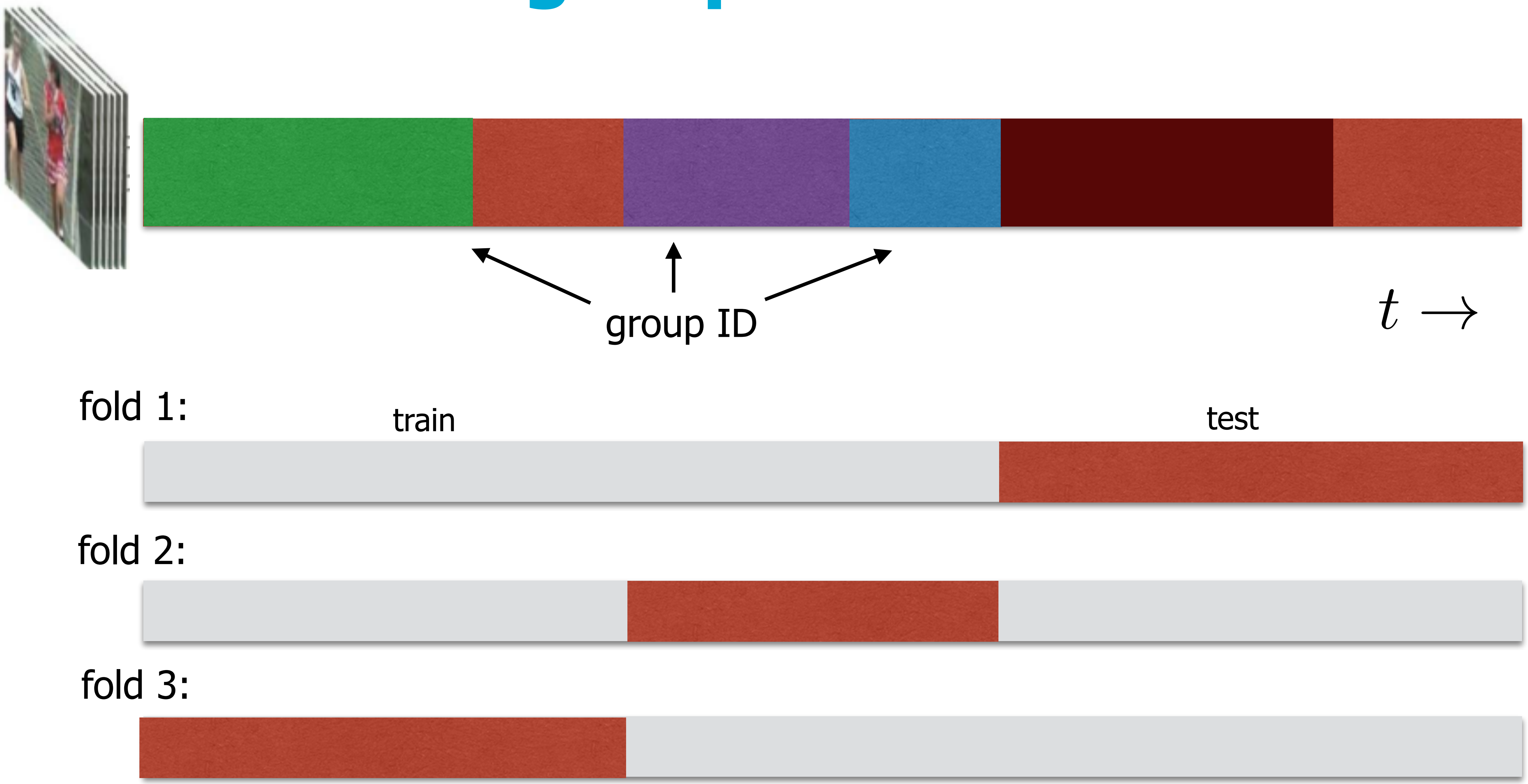
Leave-one-group-out cross-validation

- For instance: data from different persons
  data from different videos
  data from sequences



TU Delft

Leave-one-group-out cross-validation



- If you would use a random sample: will generalisation error increase/decrease/stay the same?

TU Delft

Comparing two methods

- Assume, two methods are evaluated using 10-fold cross-validation:

	fold 1	fold 2	fold 3	fold 4	fold 5	fold 6	fold 7	fold 8	fold 9	fold 10
Method 1	2.1	19.1	11.0	4.2	10.6	2.8	12.9	9.1	10.2	11.8
Method 2	4.0	19.3	10.6	5.1	10.8	4.2	12.1	11.0	11.5	11.7

- Which method is better?

Paired sample t-test

- Assume, two methods are evaluated using 10-fold cross-validation:

	fold 1	fold 2	fold 3	fold 4	fold 5	fold 6	fold 7	fold 8	fold 9	fold 10
Method 1	2.1	19.1	11.0	4.2	10.6	2.8	12.9	9.1	10.2	11.8
Method 2	4.0	19.3	10.6	5.1	10.8	4.2	12.1	11.0	11.5	11.7

- Which method is better?

	mean (std)
Method 1	9.4 (5.2)
Method 2	10.0 (4.6)

- Is this significant?

Paired sample t-test

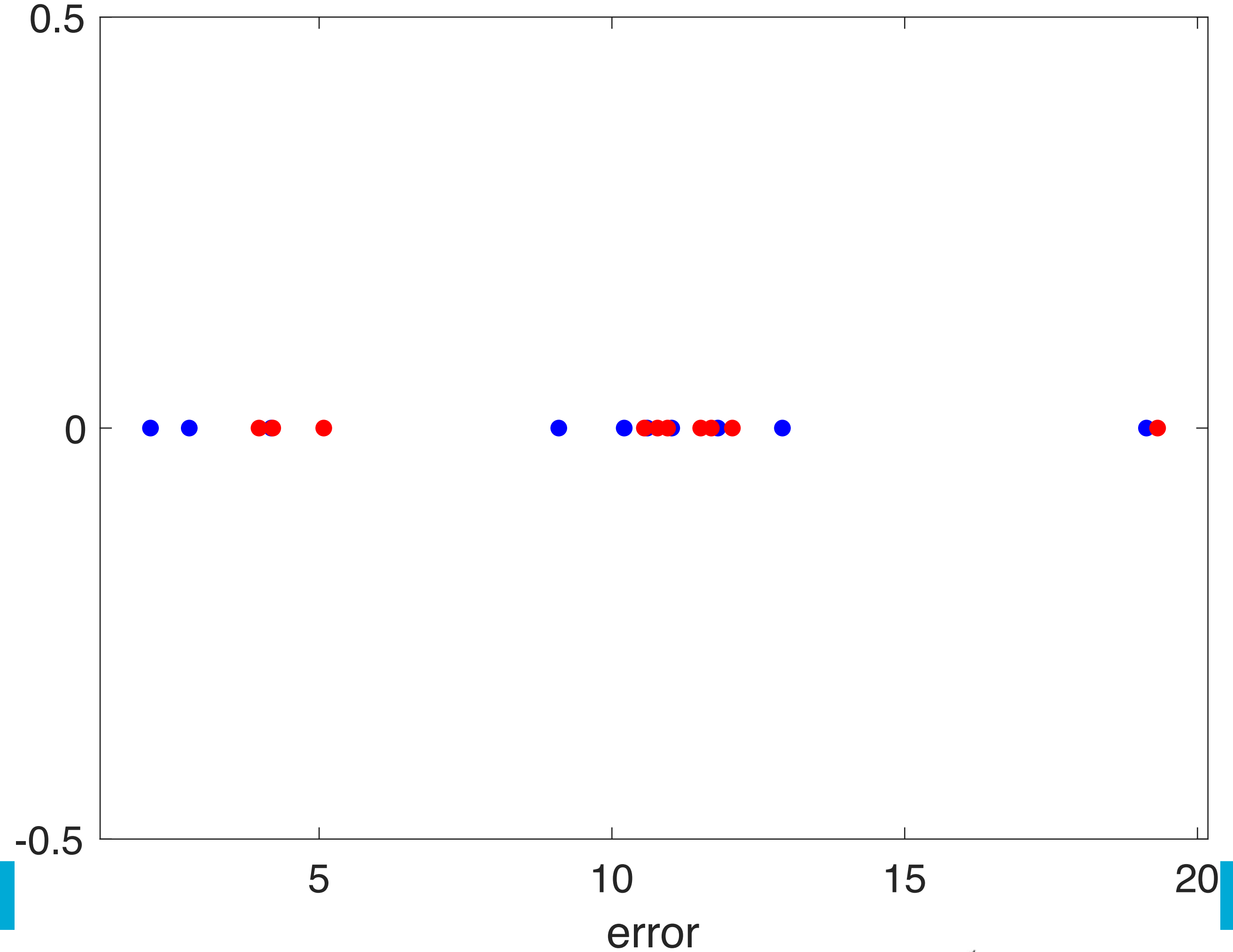
- Assume, two methods are evaluated using 10-fold cross-validation:

	fold 1	fold 2	fold 3	fold 4	fold 5	fold 6	fold 7	fold 8	fold 9	fold 10
Method 1	2.1	19.1	11.0	4.2	10.6	2.8	12.9	9.1	10.2	11.8
Method 2	4.0	19.3	10.6	5.1	10.8	4.2	12.1	11.0	11.5	11.7

- Which method is better?

	mean (std)
Method 1	9.4 (5.2)
Method 2	10.0 (4.6)

- Is this significant?



Paired sample t-test

- If results come from the same data (identical folds): look at the error **differences**:

	fold 1	fold 2	fold 3	fold 4	fold 5	fold 6	fold 7	fold 8	fold 9	fold 10
Difference	-1.9	-0.2	0.5	-0.9	-0.2	-1.4	0.9	-1.9	-1.3	0.1

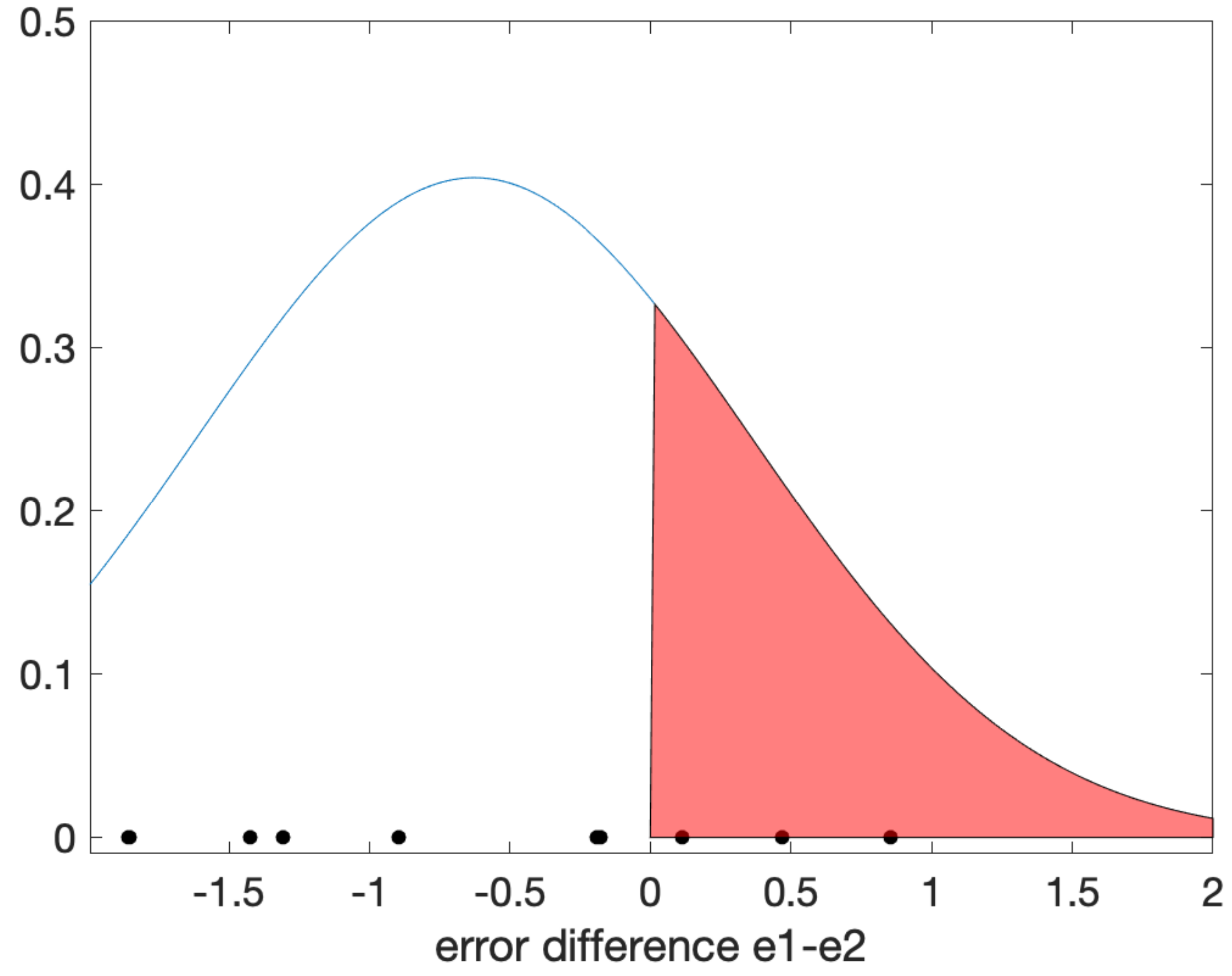
- Are these values significantly different from 0?

Paired sample t-test

- If results come from the same data (identical folds): look at the error **differences**:

	fold 1	fold 2	fold 3	fold 4	fold 5	fold 6	fold 7	fold 8	fold 9	fold 10
Difference	-1.9	-0.2	0.5	-0.9	-0.2	-1.4	0.9	-1.9	-1.3	0.1

- Are these values significantly different from 0?



TU Delft

Paired sample t-test

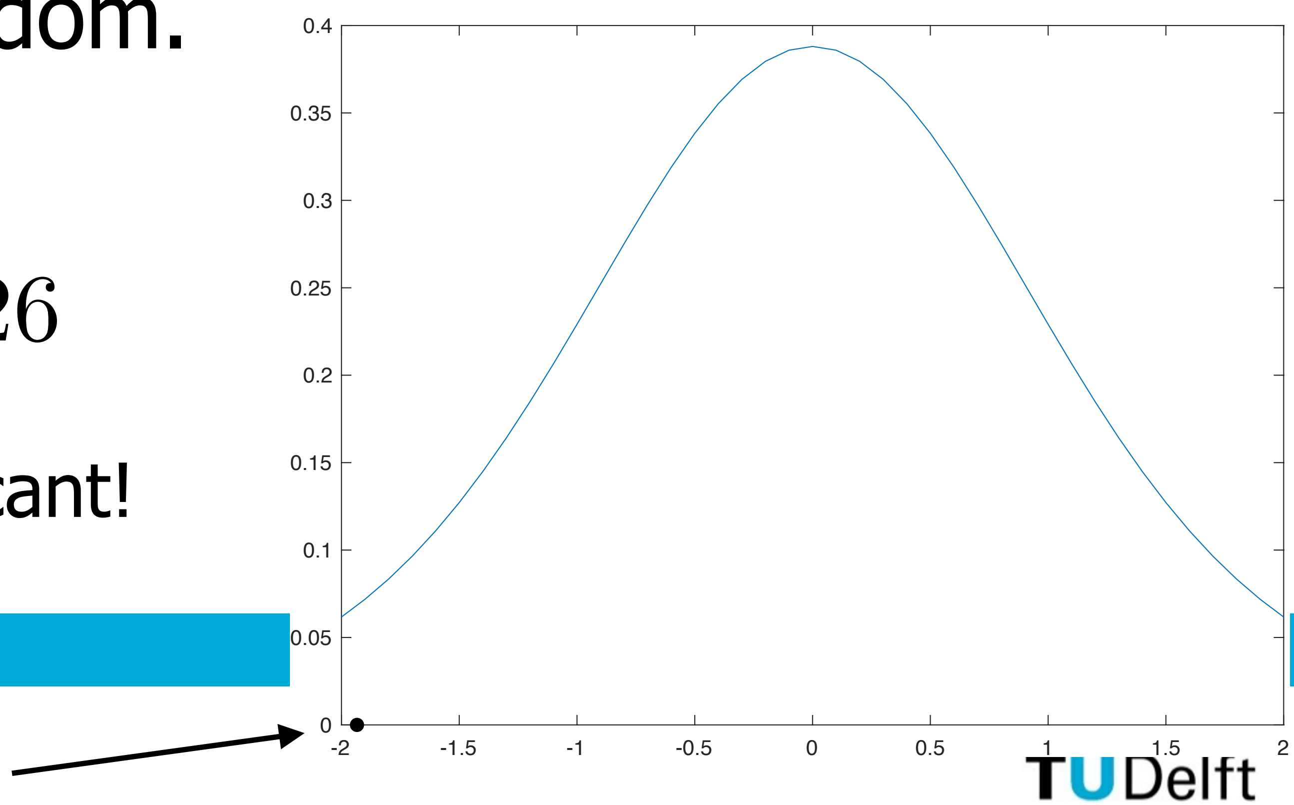
- Are the averaged errors the same?
  =
  is the difference between the averaged errors zero?
- Need a test-statistic
- You can show that the variable

$$T = \frac{\bar{e_1} - \bar{e_2}}{\sigma_{e_1 - e_2}/\sqrt{k}}$$

  has a Student-t distribution with (k-1) degrees of freedom.
- For us: T=-1.93

$$P(T \leq -1.93) = 0.0426$$

Just significant!



TU Delft

Optimisation of hyperparameters

- Machine learning methods often have ‘hyperparameters’

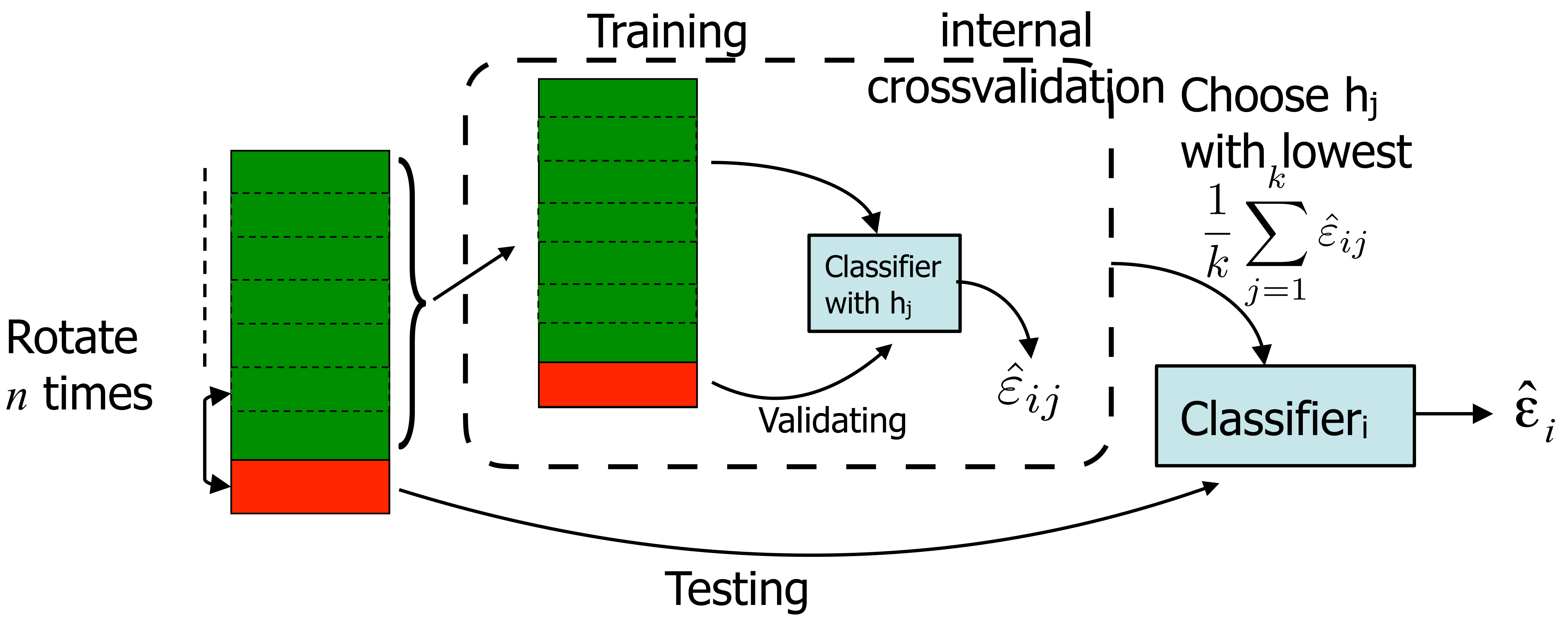
- Parzen density estimator: width parameter h
- k-nearest neighbour: number of neighbours k
- Decision trees: pruning method, stopping criterion
- Neural networks: architecture, learning rate, initialisation, batch size, regularisation, optimiser, …

- DON’T optimise these numbers by looking at the test set!
  Then you’re **CHEATING**!

TU Delft

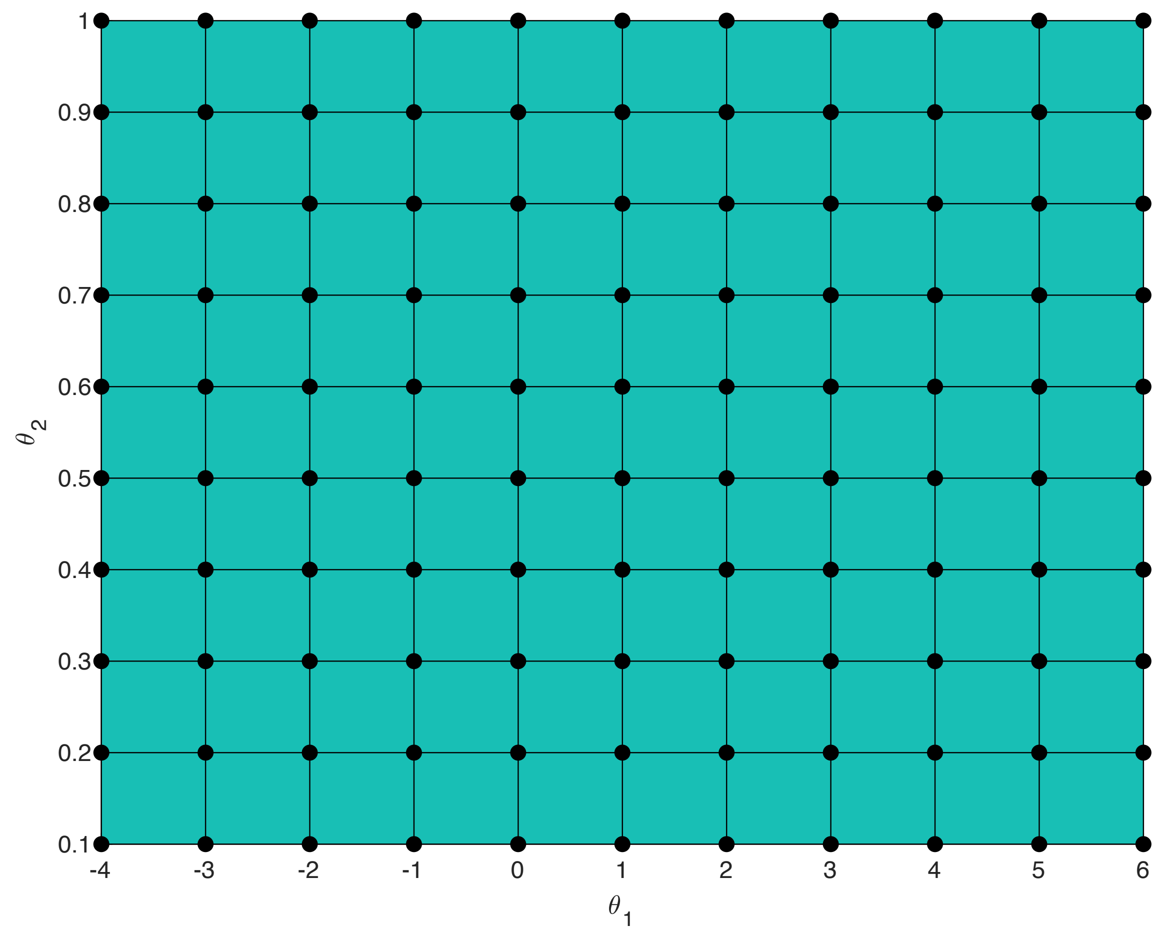
Double cross-validation

- To optimise over the hyperparameter $\{h_1, ..., h_M\}$, do cross-validation **inside** another cross-validation:



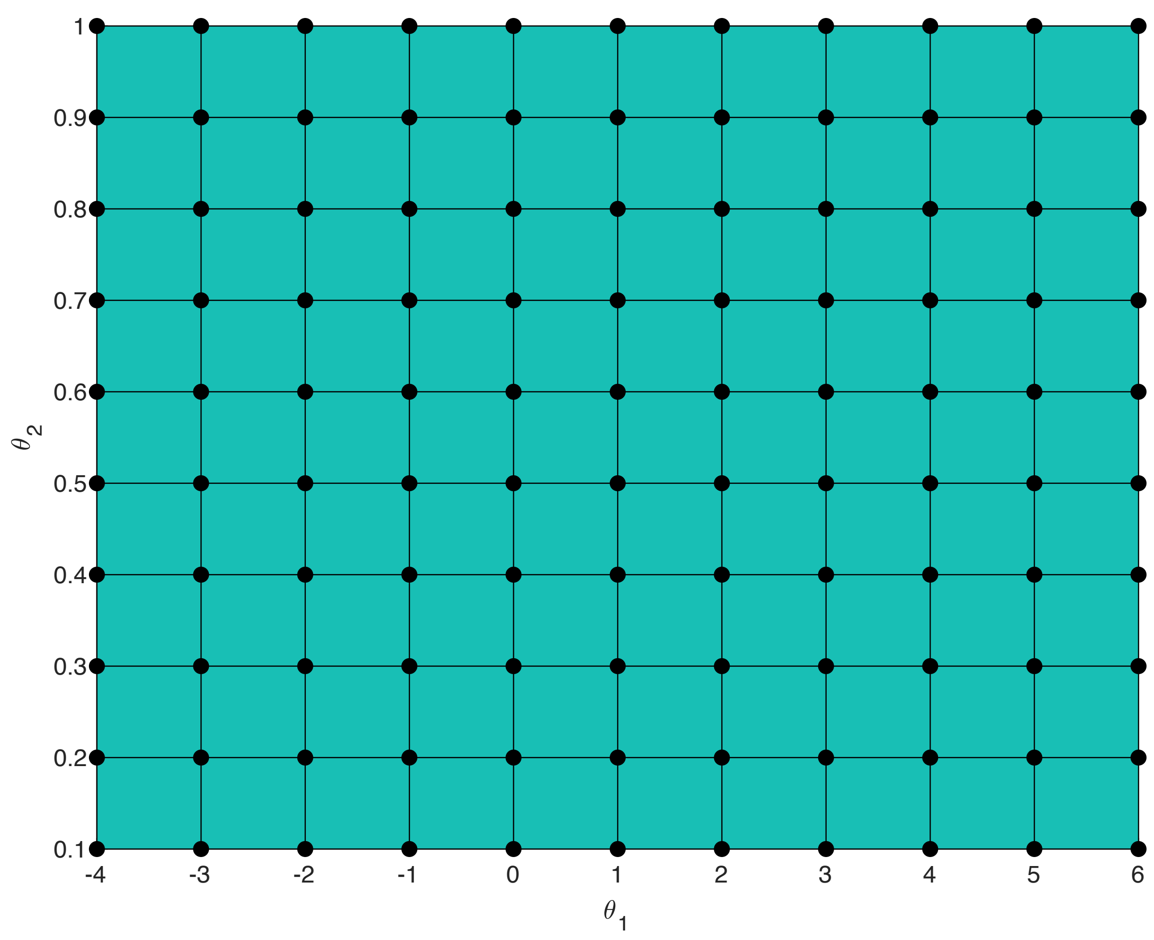
TUDelft

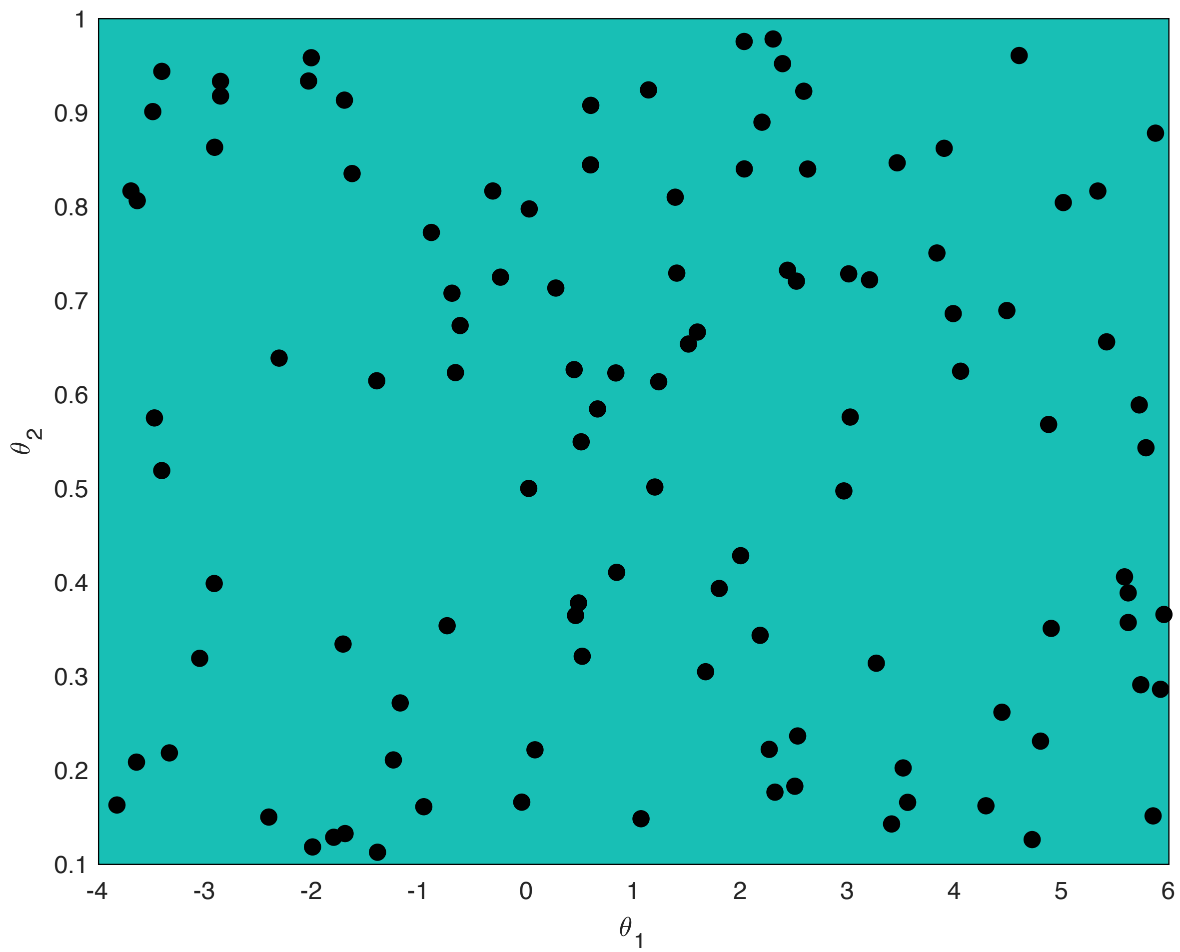
Optimisation of hyperparameters



- Grid search for 1 hyperparameter is fine
- For 2 parameters: Ok-ish, still do-able
- More than 2: Bayesian optimisation

Optimisation of hyperparameters



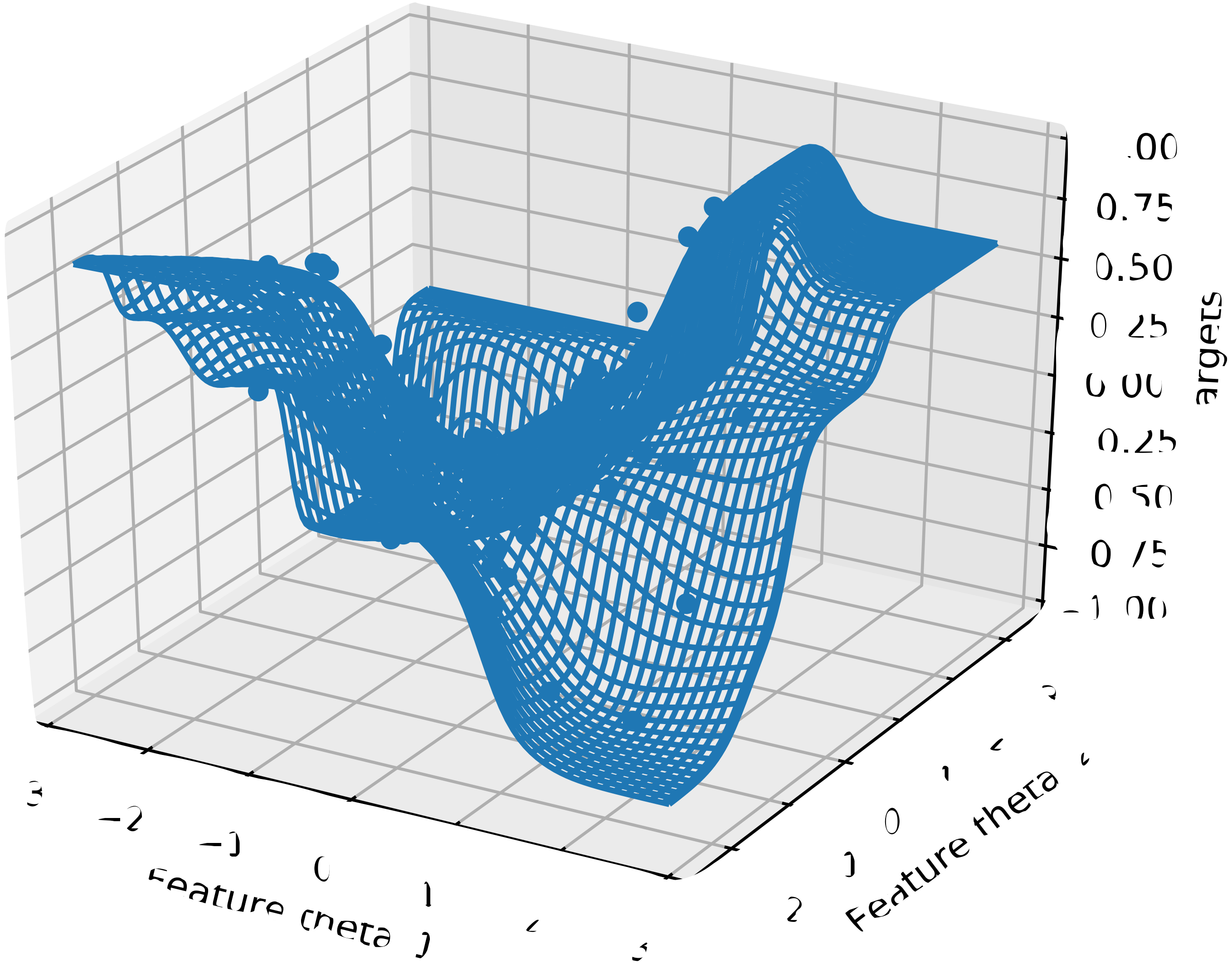


- More than 2: Bayesian optimisation
- Introduce more variability in the values of the hyperparameter $\{h_1, ..., h_M\}$

TU Delft

Bayesian optimisation

- When a new (random) hyperparameter settings are evaluated:
- fit a Gaussian Process regression
- THEN: Find minimum of the loss
- OR: Find maximum of the uncertainty, and evaluate that set of hyperparameters



TU Delft

After finding the minimum...

- The minimum error often not the most interesting (although often this is the 'proof' in articles)

- Try to understand the advantages/distadvantages:
  - What errors are made? (inspect objects, inspect labels)
  - What classes are problematic? (confusion matrix)
  - Does adding training data help? (learning curve)
  - How robust is the model?

Reporting results

- Note, typically there's quite some noise involved in the experiments (split train-test, random initialisation,…)
- Experiments are typically repeated

- Do **NOT** just present a point estimate:

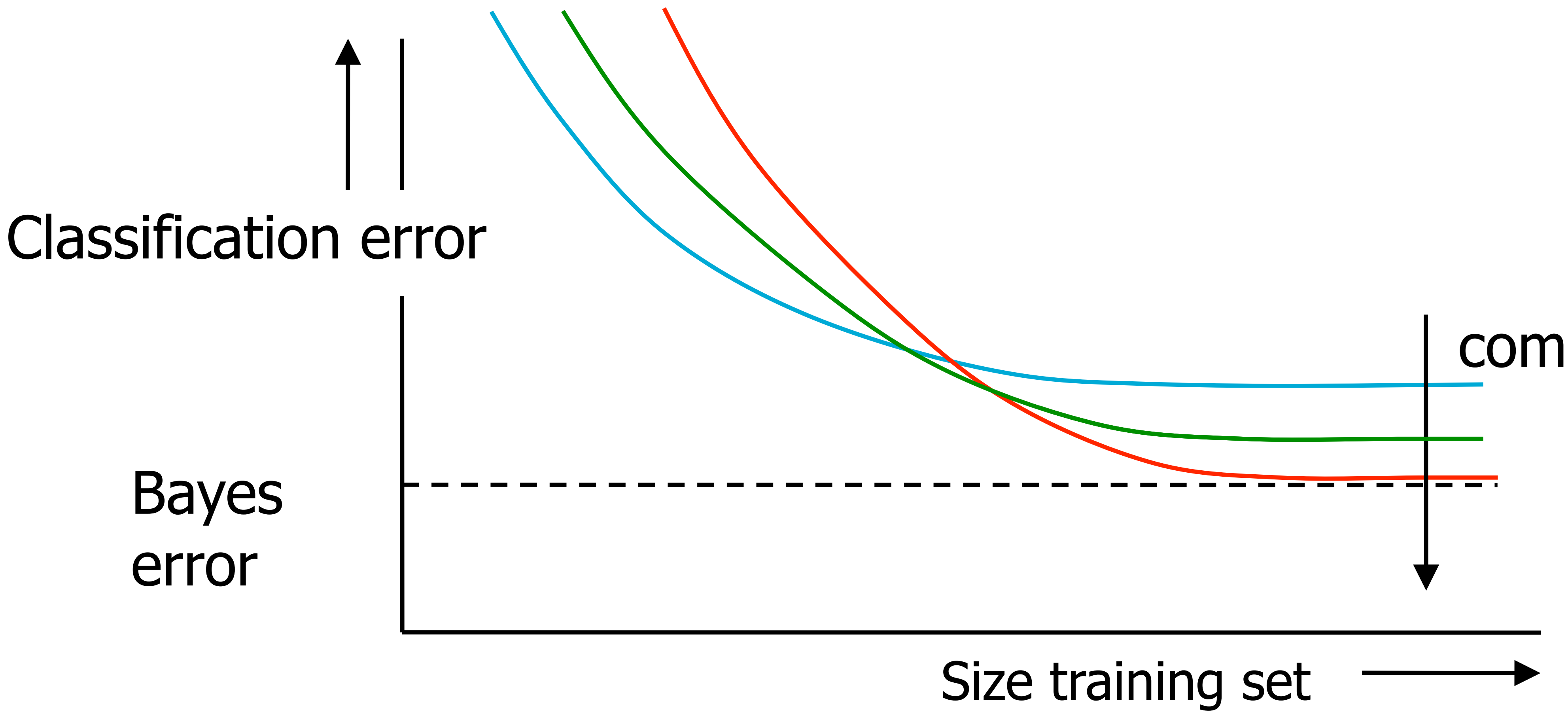
$$\begin{array}{ll} \text{Method 1} & 9.4 \\ \text{Method 2} & 10.0 \end{array}$$

- But give some idea about standard deviations:

$$\begin{array}{ll} \text{Method 1} & 9.4\ (5.2) \\ \text{Method 2} & 10.0\ (4.6) \end{array}$$

Different Classifier Complexity

- Don't claim to have the overall 'optimal' classifier
- Point out: what is good?
- What is bad?



TU Delft

Be critical!

- AI/ML/Deep Learning is a hype
- Do not trust all reported results
- Do not trust your own results

- ‘Bold numbers’ is not the goal of science:
  know what the strengths and weaknesses of a model are!
- If some method is very good, there should be disadvantages

TU Delft

Conclusions

- Many possible Machine Learning methods, all with their strengths and weaknesses
- There is no overall best classifier
- For good generalisation, the bias of the model should fit the data (distribution)

- Still interested? Consider a Graduation project in ML

- Feedback on the course; wait for an upcoming Brightspace announcement