Supervised Learning: Overfitting

Mirco Schönfeld
mirco.schoenfeld@uni-bayreuth.de

What if…

Prepare data

Chose classificator

Train it

Test it

Validate it

Results do not look good

Repeat

What‘s the problem?

Repeated testing leads to overfitting

Once the validation data is used, do not go back to improve classification!

Beware of Overfitting

Two types of classification errors:

1. Training error – misclassification on training data
2. Generalization error – expected error on *unseen* data

Overfitting:

Good results on training data (low training error) and
bad results with test/validation data (high generalization error)

Error significantly *underestimated* – severe problem in application scenarios

Detecting overfitting:

Evaluation of training with *new* data – NOT using training data!

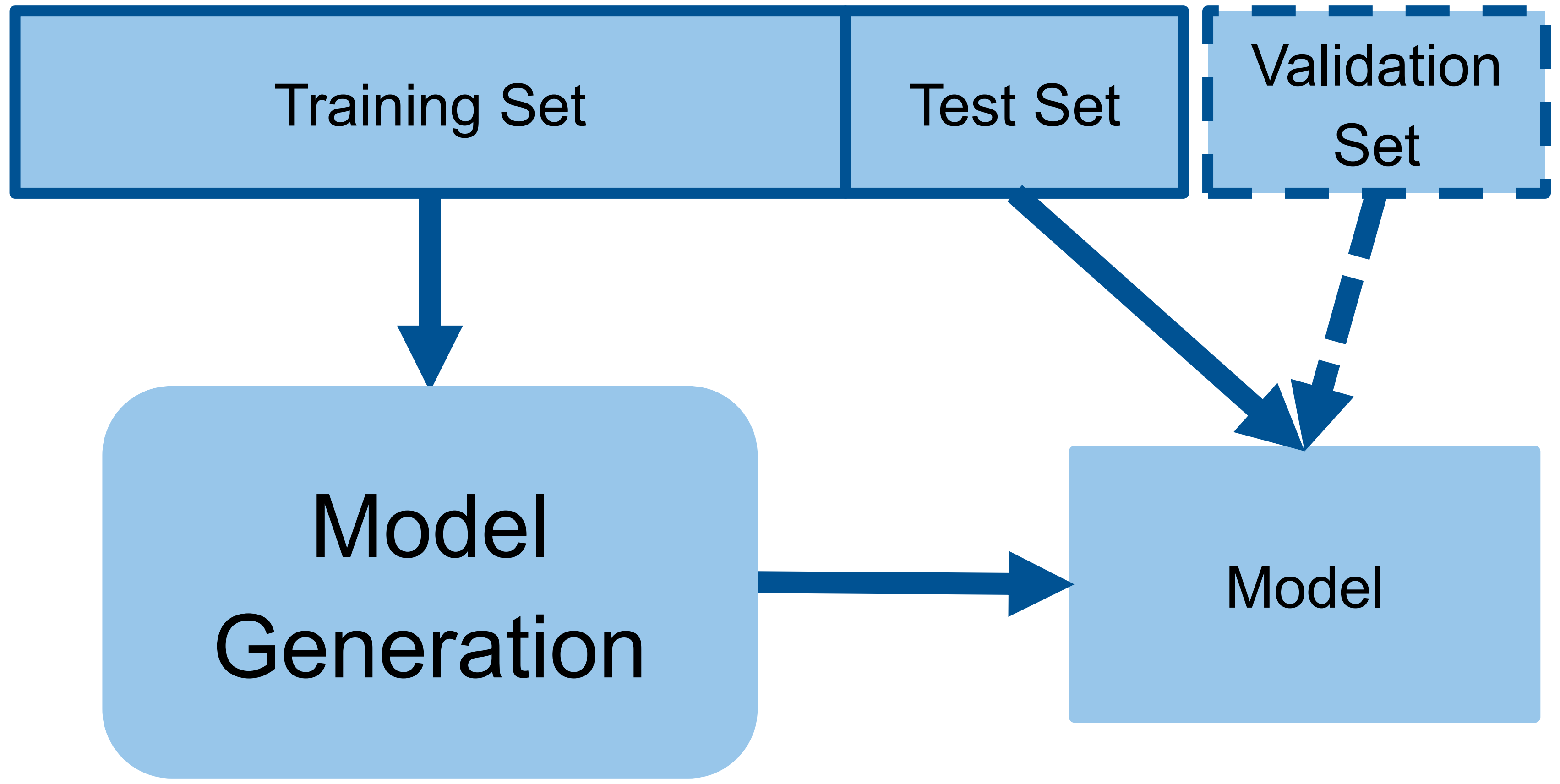
Training Test Split

Split training data *at least* in training and testing (Popular splits: 2:1 / 90:10)

Recommended: Split data in training, testing *and validation*
Splits: 80:10:10

Choose best classificator *only* on training and test data
Estimate accuracy & tune parameters of model

Keep validation-data *secret*! Use that only *once* to estimate the generalization power of the model!



Terminology of test and validation data is often mixed up.

Cross-Validation

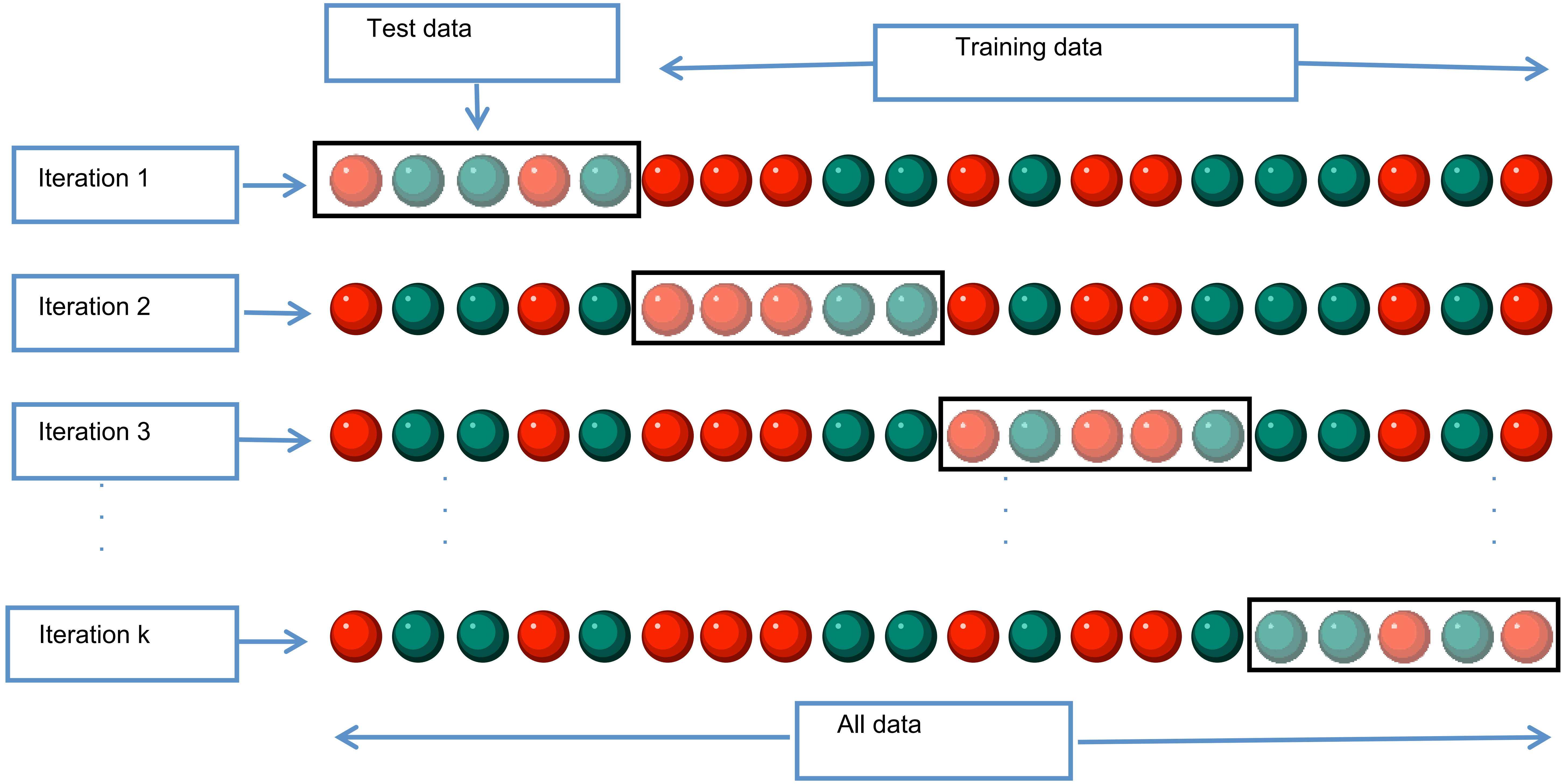
Cross-validation is a technique to help choosing classificator and optimal parameters

Partition data in *k* non-overlapping parts of equal size

During *i*th iteration, use data in partition D_i for validation, all other data as training data

Quality of classificator:

mean over all *k* iterations



https://en.wikipedia.org/wiki/File:K-fold_cross_validation_EN.svg

Exhaustive Cross-Validation

Cross-validation methods which learn and test on all possible combinations to divide the original sample into training and test set.

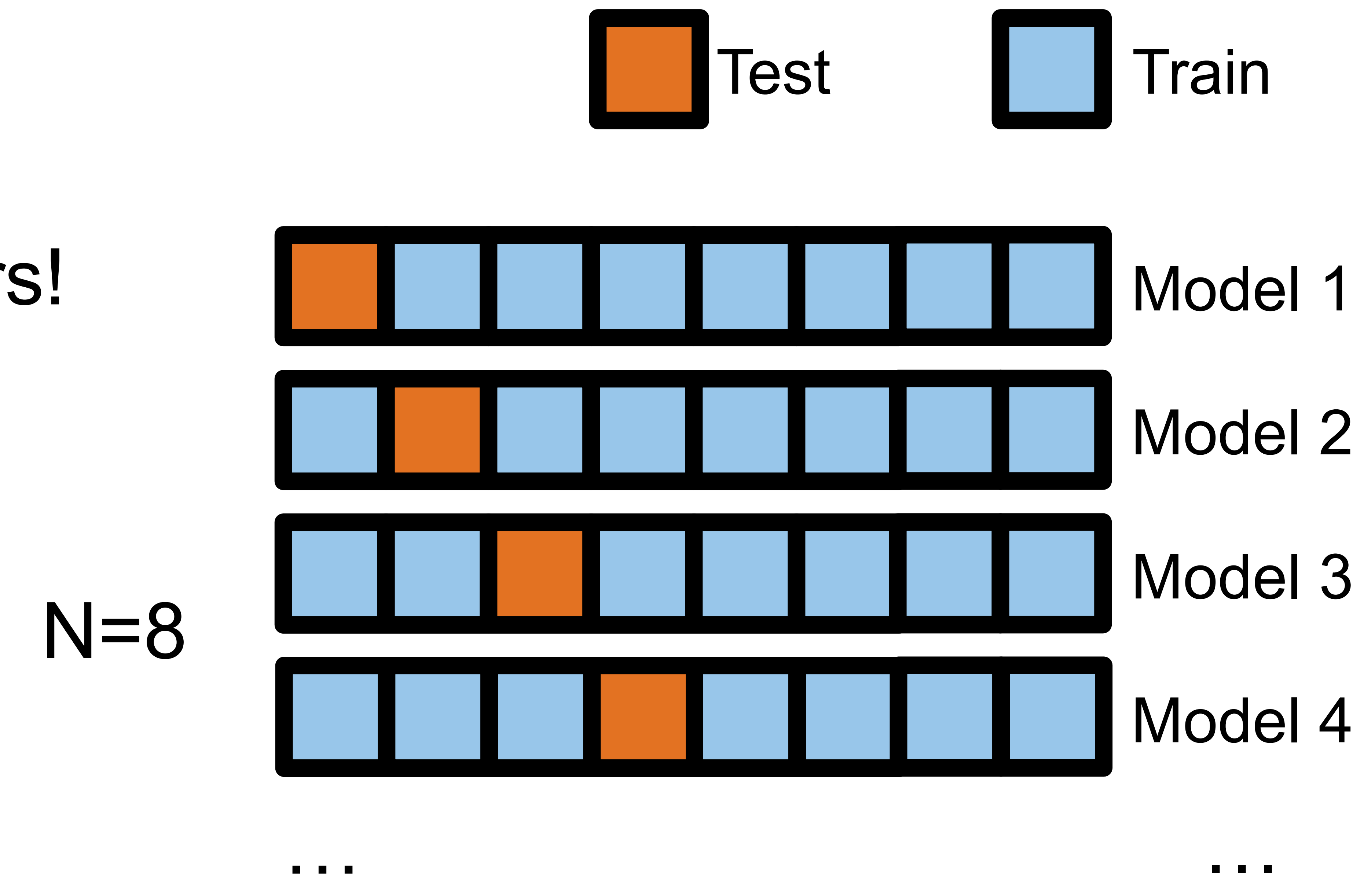
Leave-p-out cross-validation:

Use p observations as the test set and the remaining observations as training set.

Leave-one-out cross-validation:

Leave-p-out cross validation with p=1

Means finding one classificator for each instance – N classificators!



Imbalanced data

Imbalance:

Number of samples of different classes are diverging significantly.

Often, collecting samples of a certain class is difficult because these are rare events.

Consequences of building models using imbalanced data:

- Bias

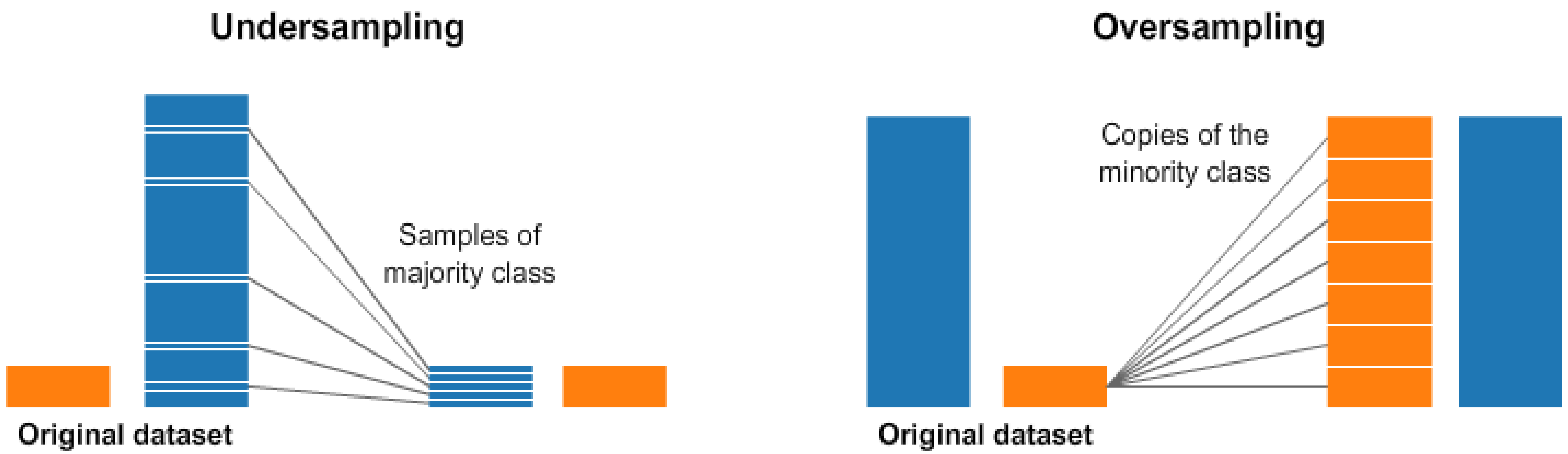
  Classifiers are more sensitive to detecting the majority class
- Optimization metrics

  Metrics like accuracy may not report true performance

Has implications for sampling for cross validation!

Resampling

Balancing classes by removing samples from the majority class (under-sampling) and/or adding more examples from the minority class (over-sampling)



Various strategies, e.g. under-sampling by generating cluster-cendroids, over-sampling by synthesizing elements (SMOTE), …

https://www.kaggle.com/rafjaa/resampling-strategies-for-imbalanced-datasets

Overfitting can occur on subtle ways

- Evaluation and task are adapted to solution
- Preprocessing of data tells something about solution
- Unbalanced data
- Little variety of data
- New observations
- Insufficient data

No Free Lunch Theorem

Choosing an appropriate algorithm requires making *assumptions*

With no assumptions, there will be no universal algorithm „better“ than random choice

> […] what an algorithm gains in performance on one class of problems is necessarily offset by its performance on the remaining problems;

Wolpert, David H., and William G. Macready. "No free lunch theorems for optimization." IEEE transactions on evolutionary computation 1.1 (1997): 67-82.