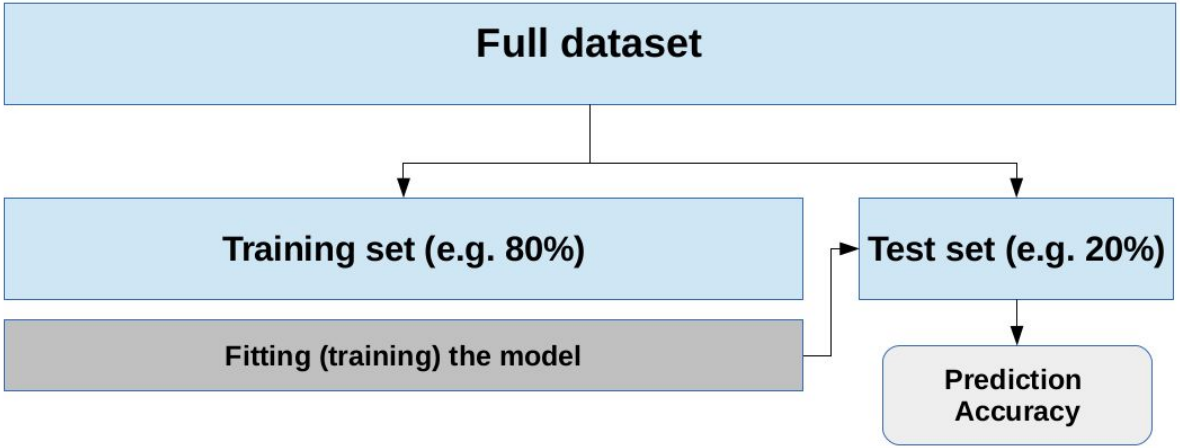
Resampling methods

Filippo Biscarini (CNR, Milan, Italy)

filippo.biscarini@cnr.it

Training and test sets

Sampling the training and the test sets



- To correctly assess the performance of a predictive model we measure it on independent data → test data
- However we can sample many different training and test sets!

Resampling the data

- Resampling involves **repeatedly sampling** the training and test datasets: each time, the model is **refitted** in the training set and **evaluated** in the test set
- You can e.g. estimate the **variability** of a predictive model or the effect of modifying the model or method:
    - **Model assessment**
    - **Model selection**

Model assessment

- Resample the data to measure the predictive ability (performance) of a model
  - in a valid way (test data)
  - in a robust way (resampling → many “test” data)
- Resample to measure the variability of model performance / estimated parameter
  - cross-validation repeated n times → average value +/- std dev

Model selection

- All methods/models have some complexity degree that controls how complex the method/model is and can be tuned:
  - cross-validation to select the best value for the complexity (e.g. the lowest error / highest accuracy)
  - the best model is chosen and used for the final analysis (applied to the training set)

Resampling the data

- Several resampling methods exist
- We will examine two such methods:
    1. **validation set approach**
    2. **cross-validation**

[validation set ~ test set]

The validation set approach

training set | **validation set**

- We split the data in **two random subsets**: training and validation (test)
- 10%/90%, 20%/80%, 30%/70% etc.
- This is what we already did!
- Repeat this *n times* and you get **robust estimates** of the model performance

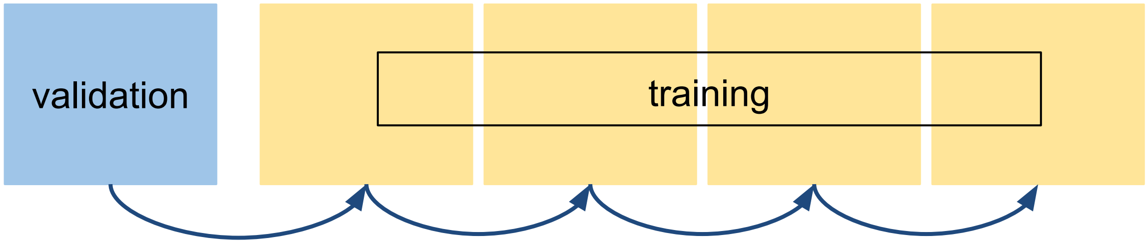
The validation set approach

training set | **validation set**

Drawbacks:

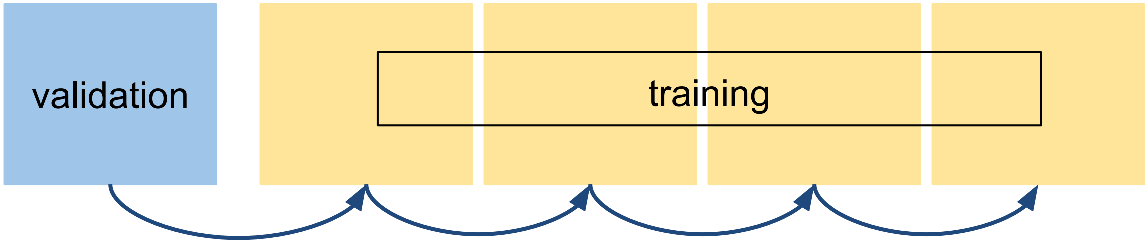
- **highly variable** (depending on the random partition of the data)
- only a subset of the data is used to train (fit) the model → **potentially underestimate model performance**

k-fold cross-validation



- *k* random partitions of equal size
- each partition in turn is used for validation, the rest for training
- *k* estimates of model performance

k-fold cross-validation



- *k* random partitions of equal size
- each partition in turn is used for validation, the rest for training
- ***k* estimates** of model performance

$$CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} MSE_i$$

k-fold cross-validation

- Lower variability than the validation set approach
- cross-validation works well in **finding the minimum point** in the estimated test MSE curve → model selection
- In cross-validation each observation/record is used both to train the model and to test it → more data are used here than in the validation set approach → lower bias
- cross-validation is therefore expected to have **both lower variance** and **lower bias** than the validation set approach → more accurate estimate of model performance
- typical values for *k* are **k=5** and **k=10**

k-fold cross-validation

validation-set approach
k-fold cross-validation
Exercise 3.2

→ 3.training_testing.ipynb

Cross-validation: right and wrong

- Consider a **regression problem**: **100 samples**, **50,000 features** (variables, e.g. ‘omics data):

    1. Find the 50 features with the **strongest correlation** with the response variable
    2. Apply a **predictor** (e.g. multiple linear regression) with only these 50 **selected features**

Estimate the **prediction error**: can we apply cross-validation in step 2?

Cross-validation: right and wrong

- Consider a **regression problem**: **100 samples**, **50,000 features** (variables, e.g. ‘omics data):
    1. Find the 50 features with the **strongest correlation** with the **response variable**
    2. Apply a **predictor** (e.g. multiple linear regression) with only these 50 **selected features**

Estimate the **prediction error**: can we apply cross-validation in step 2? → **NO!**

Cross-validation: right and wrong

Estimate the **prediction error**: can we apply cross-validation in step 2? → **NO!**

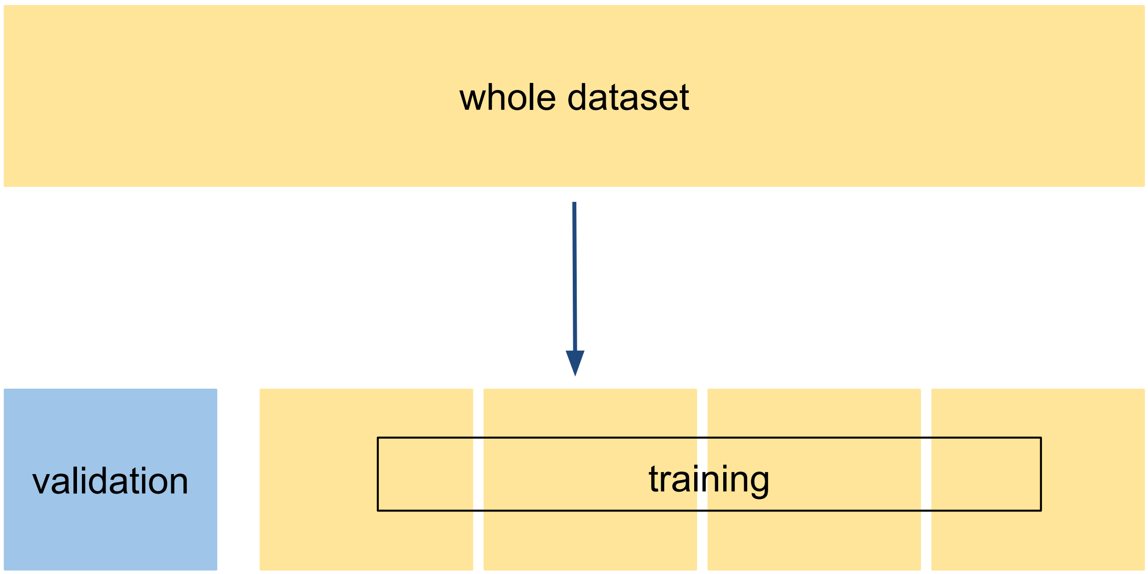
- in Step 1, the **model has already used the response** of the training data
- Features have been **"cherry picked"** based on the data: this is already **training**, and the correlation with the response may be a result of the specific configuration of this dataset (a "quirk" in the data)

Cross-validation: right and wrong

Estimate the **prediction error**: can we apply cross-validation in step 2? → **NO!**

- **Wrong!** → select variables on the whole dataset, then apply cross-validation
- **Right!** → first split the data in training and test sets, then select variables (part of training)

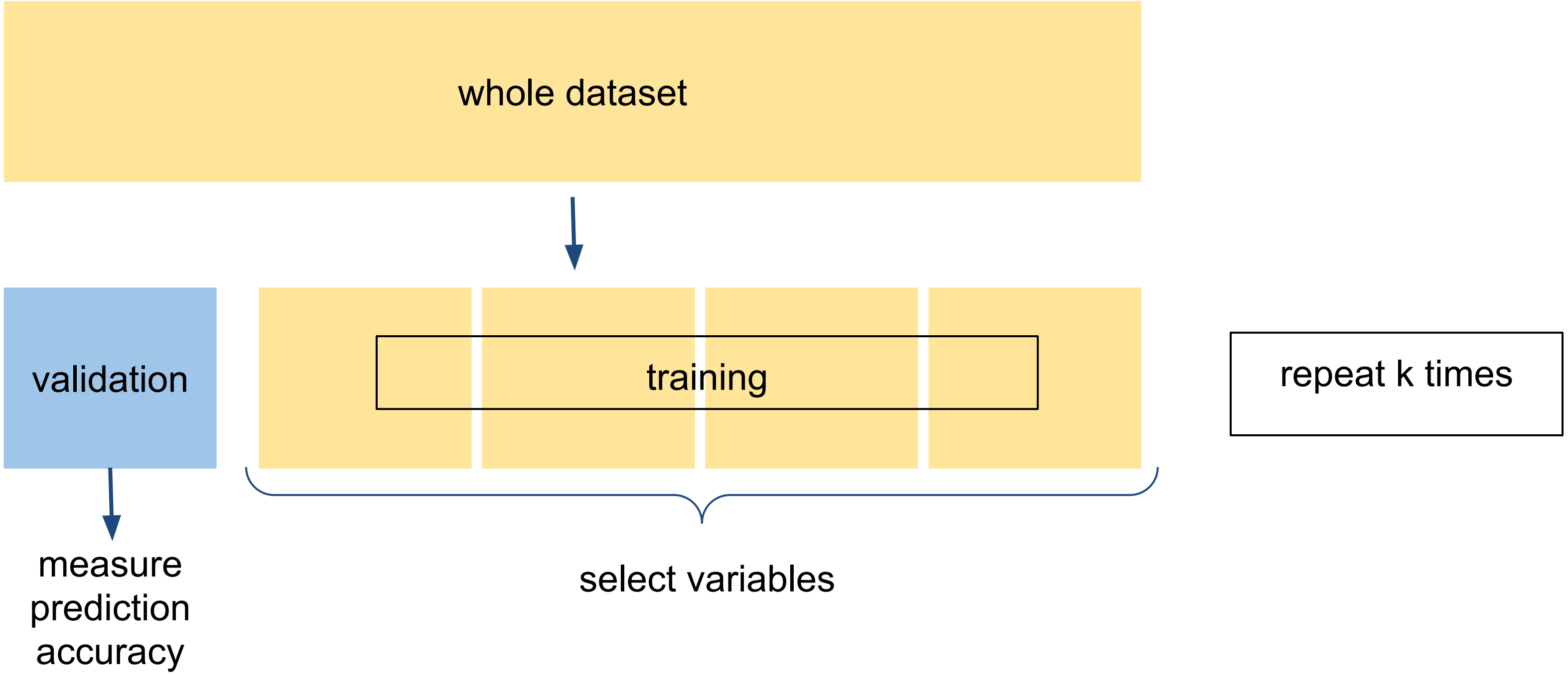
Cross-validation: wrong way



select variables (based on the response)

measure prediction accuracy

Cross-validation: right way



Cross-validation: right way