Data Mining

Predictive Modelling

Evaluation Methodologies, Model Selection and Comparison of Models

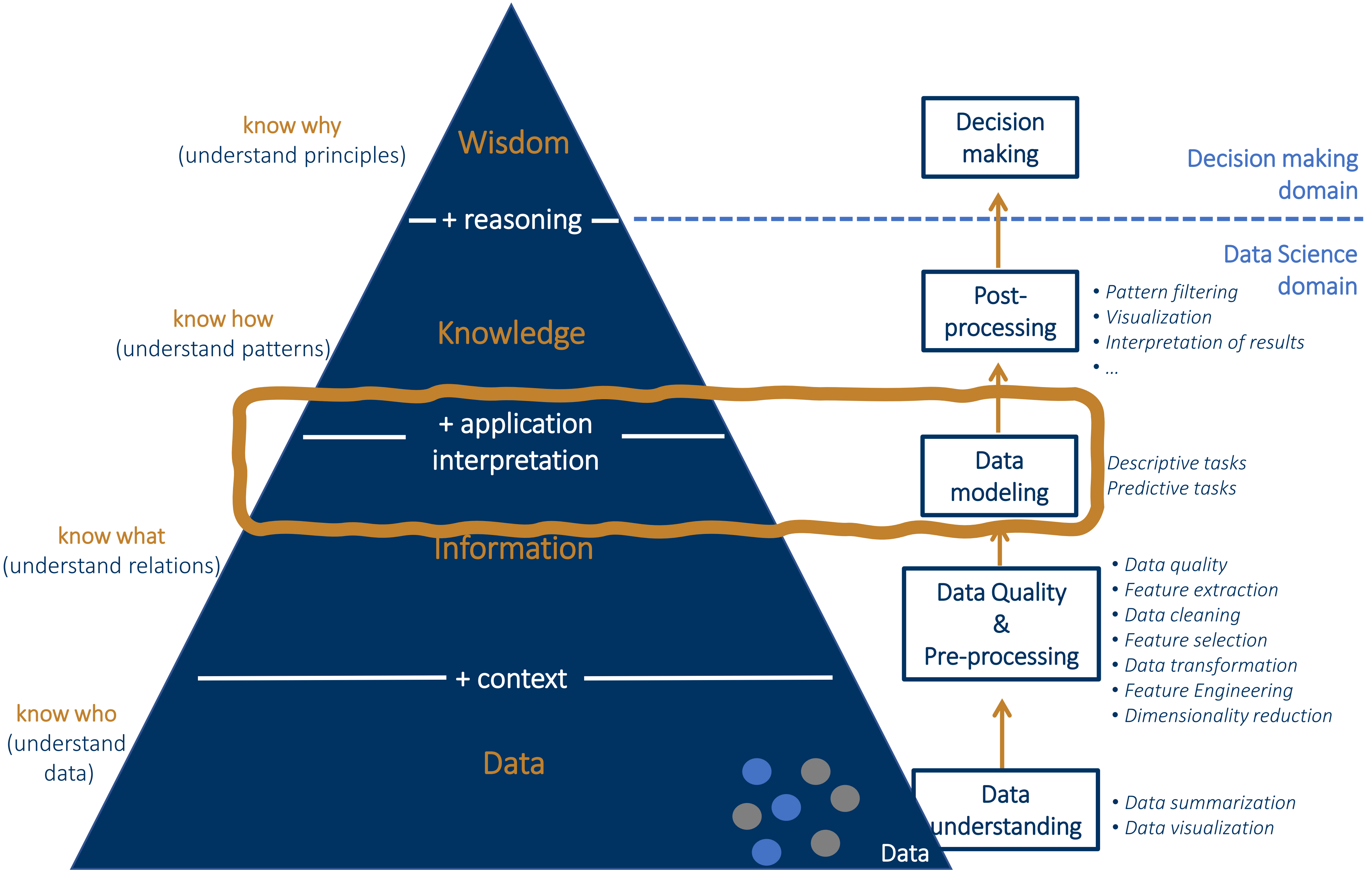
Raquel Sebastião

Departamento de Eletrónica, Telecomunicações e Informática

Universidade de Aveiro

raquel.sebastiao@ua.pt

2022/2023


know why
(understand principles)
Wisdom
+ reasoning
know how
(understand patterns)
Knowledge
+ application
interpretation
know what
(understand relations)
Information
+ context
know who
(understand
data)
Data
Data
Decision making
Decision making domain
Data Science domain
Post-processing
• Pattern filtering
• Visualization
• Interpretation of results
• ...
Data modeling
Descriptive tasks
Predictive tasks
Data Quality & Pre-processing
• Data quality
• Feature extraction
• Data cleaning
• Feature selection
• Data transformation
• Feature Engineering
• Dimensionality reduction
Data understanding
• Data summarization
• Data visualization

Contents

- Evaluation methodologies
  - Performance estimation
  - Measures/metrics
    - confusion matrix
    - measures of performance
  - Techniques
    - Holdout
    - K-Fold Cross-Validation
    - Leave-One-Out Cross Validation
- Model selection & hyperparameter tunning
- Comparison of models
- Summary

Performance estimation

Setting

- Given a **data set** $\{(x_i, y_i)\}_{i=1}^{N}$, where each **object** is represented by a **D+1**–tuple:
  - Predictors variables: (D–dim) feature vector $\mathbf{x}_i = \left[x_i^1 \quad x_i^2 \quad \dots x_i^D\right]^T \in \mathbb{R}^D$
  - Target variable: corresponding **label** $y_i \in \boldsymbol{Y}$
- There is an **unknown** function: $\mathrm{Y} = F(\mathrm{X})$ that maps the values of a set of predictors into a target variable value (can be a classification or a regression problem)

Goal: Predictive task

- Learn the **model** that yields the best approximation of the unknown function *F*()
  - Classification problem
  - Regression problem

How to obtain a reliable estimates of the predictive performance of the learned model?

Performance estimation: resubstitution estimate

Estimate of the performance of the model by evaluating on the same data set used for learning the model

- Unreliable and should not be used as they tend to be over-optimistic!
  - Models are obtained with the goal of optimizing the selected prediction error statistic on the given data set
    - Thus, it is expected to get good scores on the data used for learning
  - The given data set is just a sample of the unknown distribution of the problem being tackled
    - Ideally: compute the performance of the model on this distribution
  - As this is usually impossible, the best we can do is to evaluate the model on new samples of this distribution

Use **test set (unseen data)** of class-labeled tuples instead of training set when assessing performance

Performance estimation

- Obtain a **reliable estimate** of the expected prediction error of a model on the unknown data distribution
- In order to be reliable it should be based on evaluation on unseen cases: test set

The golden rule

The data used for evaluating (or comparing) any models cannot be seen during model development

Contents

- Evaluation methodologies
  - Performance estimation
  - Measures/metrics
    - confusion matrix
    - measures of performance
  - Techniques
    - Holdout
    - K-Fold Cross-Validation
    - Leave-One-Out Cross Validation
- Model selection & hyperparameter tunning
- Comparison of models
- Summary

Evaluation of models: confusion matrix

- Confusion matrix for a binary classification problem

		Predicted class	
		P	*N*
True class	*P*	**TP** True Positive	**FN** False Negative
	N	**FP** False Positive	**TN** True Negative

- TP: hit
- FN: miss (type II error)
- FP: false alarm (type I error)
- TN: correct rejection

Evaluation of models: measures of performance

- Accuracy $= \dfrac{TP+TN}{TP+FN+FP+TN}$ (proportion of correct predictions)
- Error Rate = 1- Accuracy (proportion of predictions that are incorrect)
- Precision $= \dfrac{TP}{TP+FP}$ (proportion of correct positive predictions)
- Recall $= \dfrac{TP}{TP+FN}$ (proportion of positive objects correctly predicted)
  - F-measure: weighted combination of Precision and Recall

$$F_\beta = \frac{1}{\alpha \cdot \frac{1}{Precision} + (1-\alpha) \cdot \frac{1}{Recall}} = \frac{(\beta^2+1) \times Precision \times Recall}{\beta^2 Precision + Recall}$$

    - $\boldsymbol{F_1}$: β = 1 then is the harmonic mean of ***Precision*** and ***Recall***

- Area Under the Curve (AUC)

Contents

- Evaluation methodologies
    - Performance estimation
    - Measures/metrics
        - confusion matrix
        - measures of performance
    - Techniques
        - Holdout
        - K-Fold Cross-Validation
        - Leave-One-Out Cross Validation
- Model selection & hyperparameter tunning
- Comparison of models
- Summary

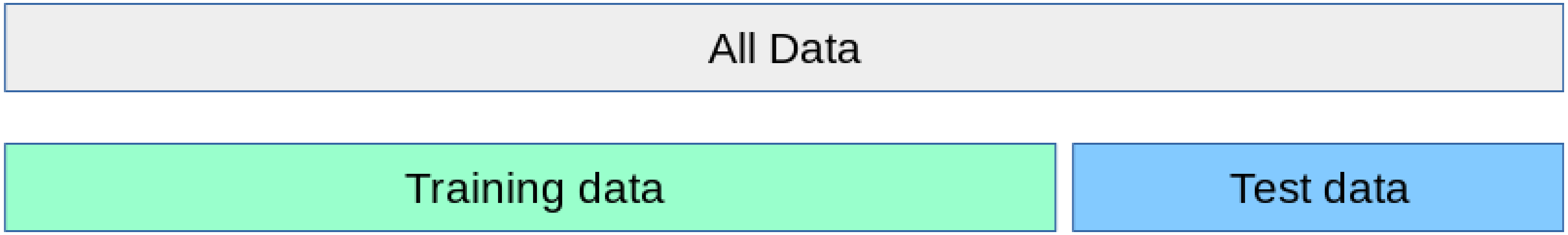
Performance estimation techniques

- Holdout
- K-Fold Cross-Validation
- Leave-One-Out Cross Validation
- Boostrap

Performance estimation techniques: Holdout

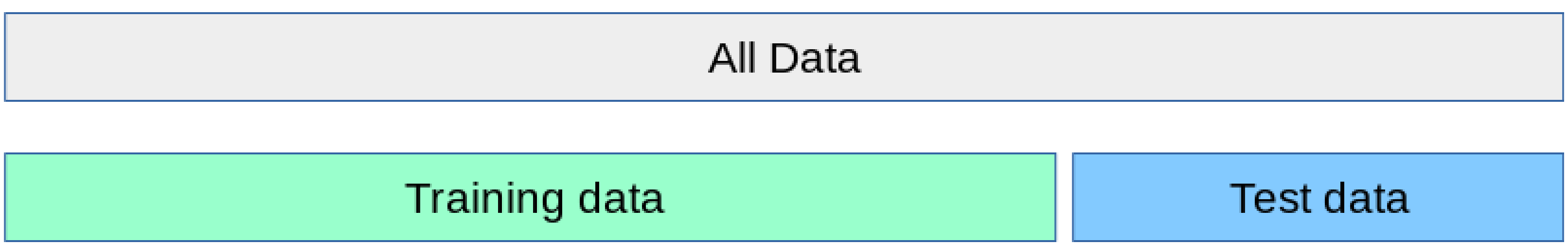
It consists of randomly dividing the available data sample in two sub-sets:

- one used for training the model
- the other for testing/evaluating it
  - a frequently used proportion is 70% for training and 30% for testing
- only one prediction error score is obtained (no average error nor standard error)

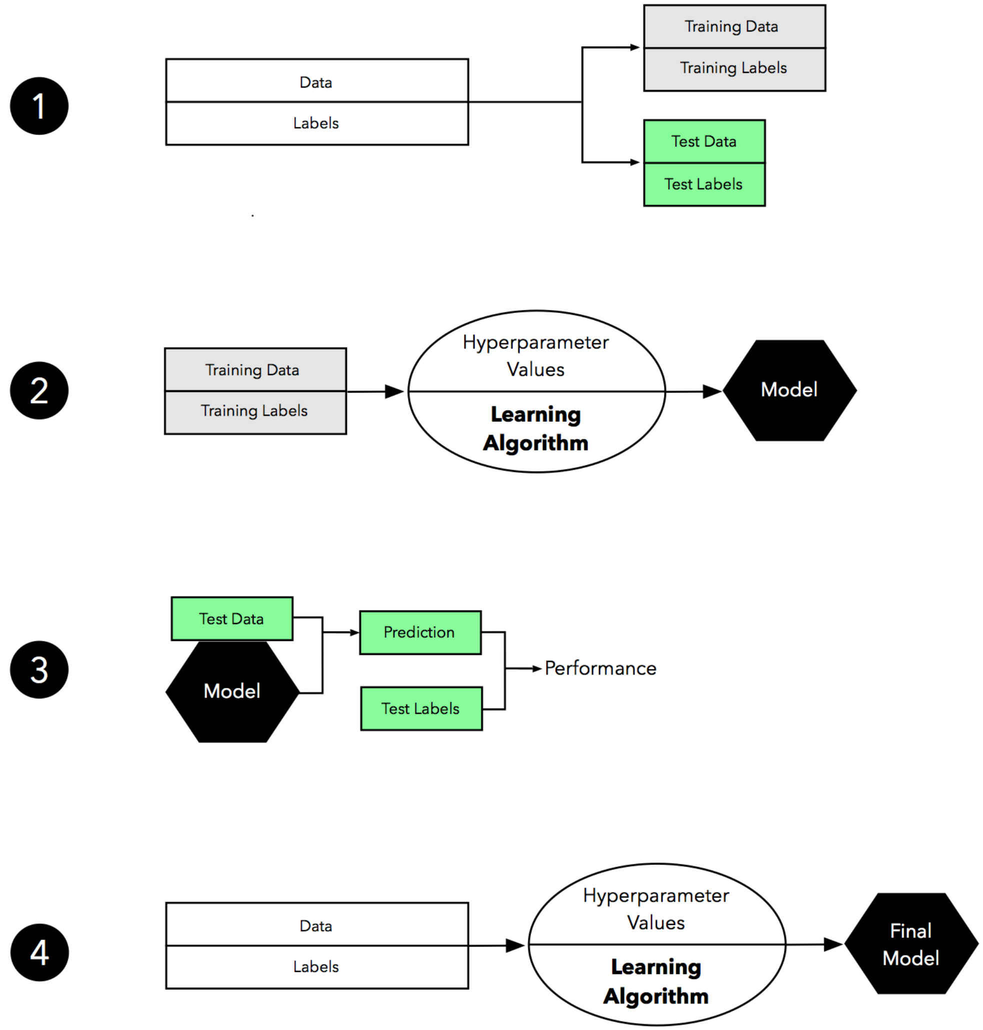


Performance estimation techniques: Holdout

- Very large dataset: preferred evaluation method
- Small dataset
  - too small test set (consequence: unreliable estimates)
  - removing too much data from the training set (worse model than what could be obtained with the available data)



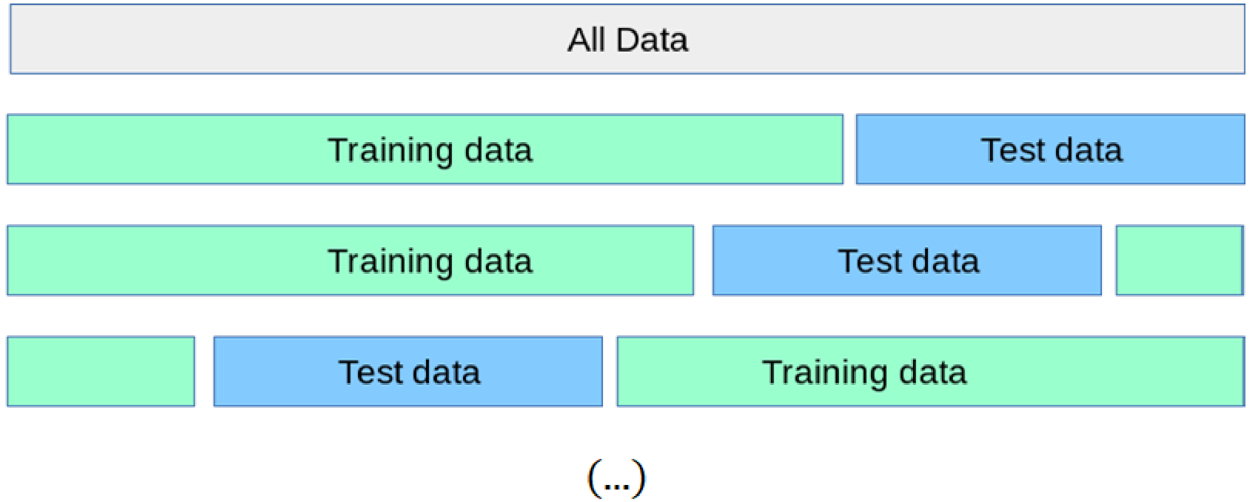
Performance estimation techniques: Holdout



Performance estimation techniques: Random subsampling

- Repeated holdout (Monte-Carlo Cross Validation)
  - the holdout process is repeated several times by randomly selecting the train and test sub-sets

- The performance is the average of different training/test
  - several prediction error scores (allow compute average error and standard error)

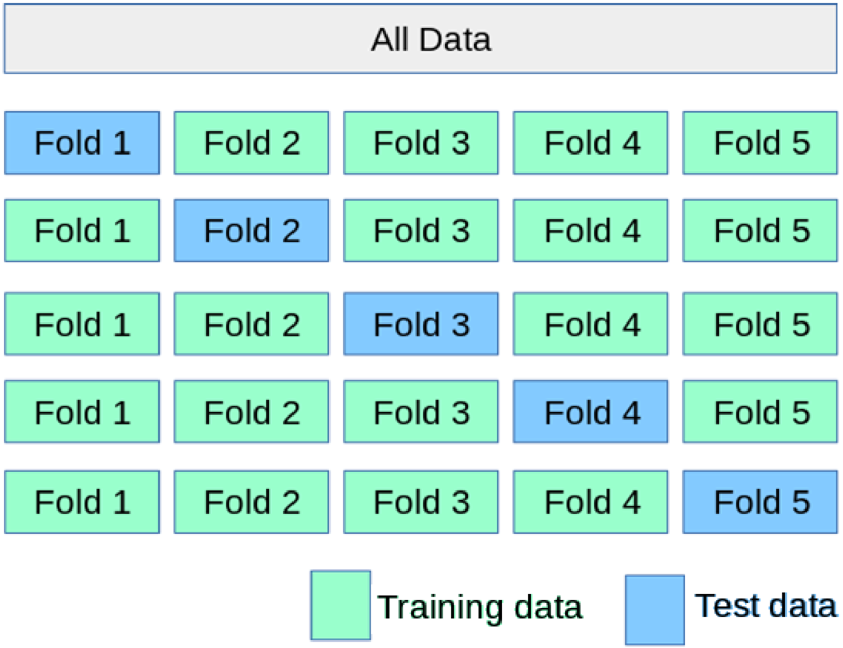


Performance estimation techniques: K-Fold Cross-Validation

Divide the data set into K partitions:

- training set with (K - 1) partitions
- test set with 1 subset

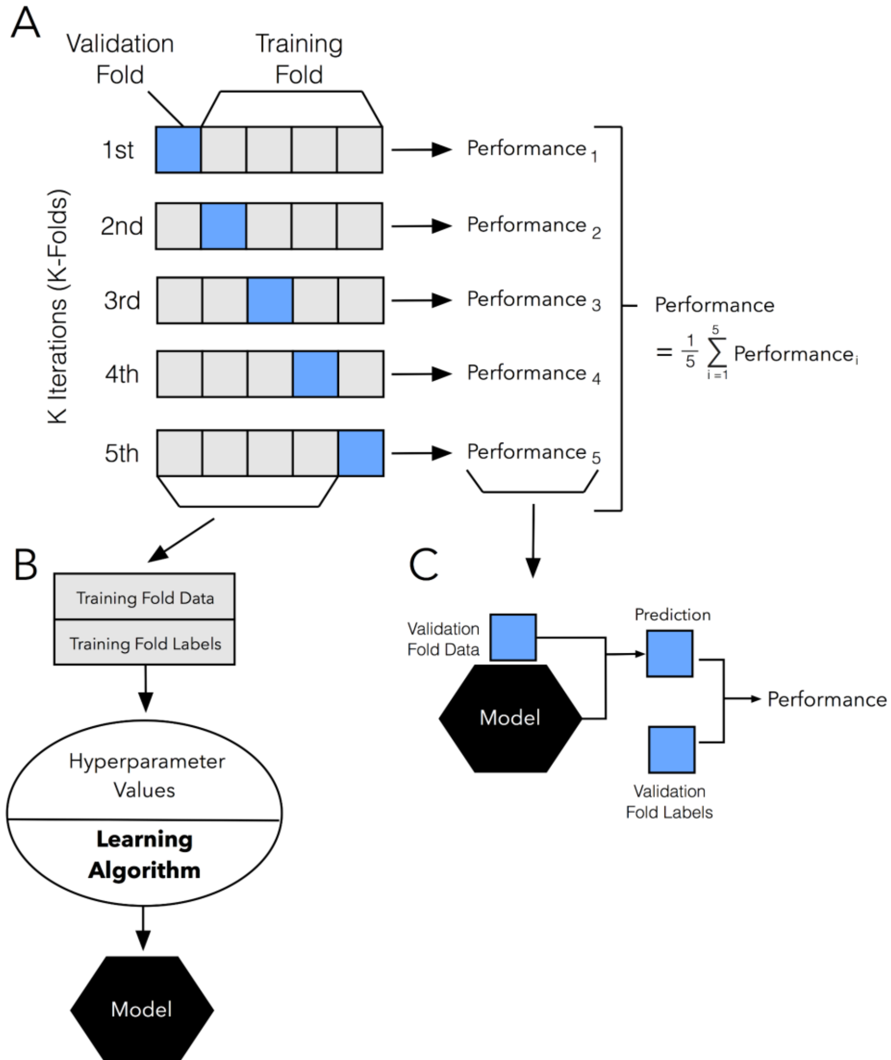
Repeat training/test K times



The performance is the average of different training/test

- several prediction error scores (allow compute average error and standard error)

Performance estimation techniques: K-Fold Cross-Validation



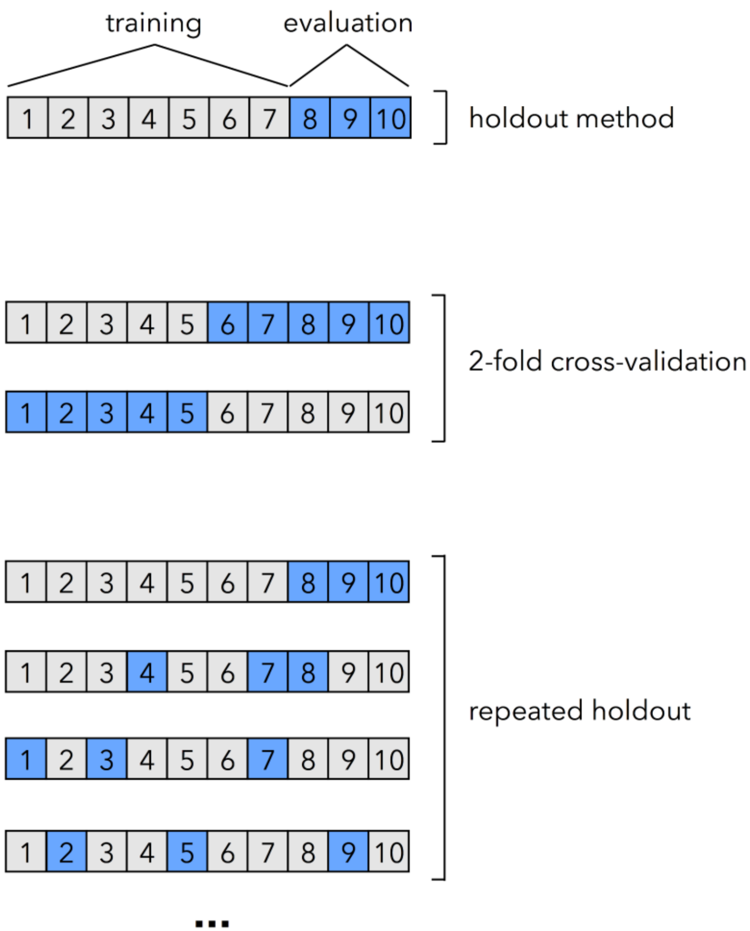
Performance estimation techniques: K-Fold Cross-Validation

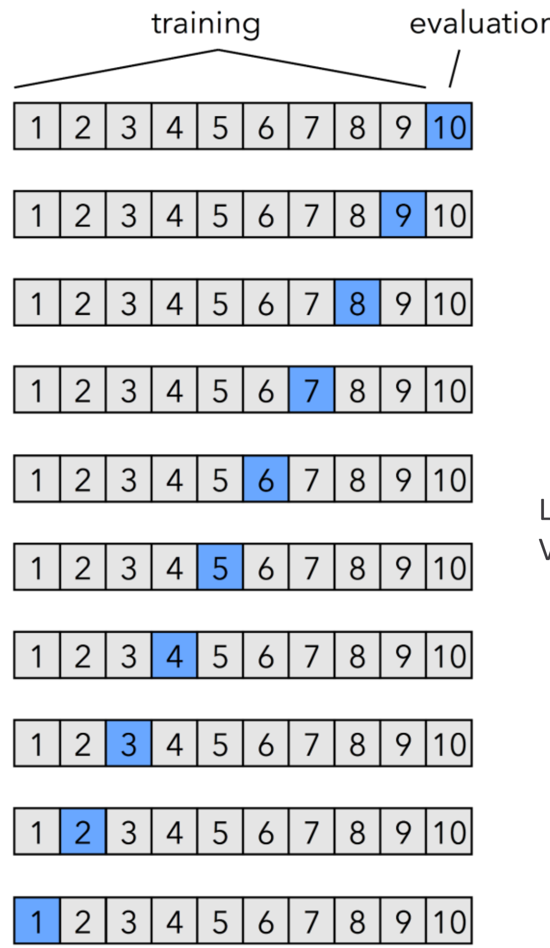
Stratified k-fold Cross Validation

- ensures that, in each fold, each class has roughly same proportion as in full data set
- if it is expected that the learning algorithm to sensitive to the target variable distribution

- **Leave One Out Cross Validation (LOOCV)**
  - n-fold CV, where n is the size of the full data set
    - in this case on each iteration, a single case is left out of the training set

Performance estimation techniques: overview





Leave-one-out Cross Validation

Contents

- Evaluation methodologies
  - Performance estimation
  - Measures/metrics
  - Techniques
- Model selection & hyperparameter tunning
  - 3-way Holdout
  - K-Fold Cross-Validation
  - Nested Cross-Validation
- Comparison of models
- Summary

Model selection & optimal hyperparameters

How can we choose the optimal hyperparameters?

Hyperparameter tuning & model selection:

- Meta-optimization task

Learning algorithm

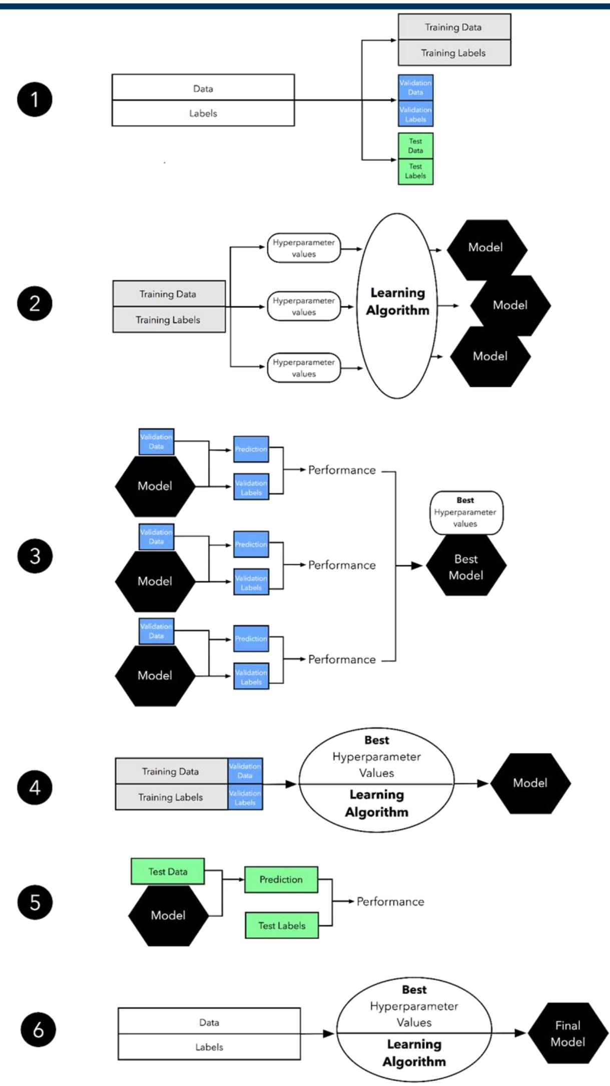
- optimizes an objective function on the training set*

Hyperparameter optimization

- optimize a performance metric
  - another task on top of optimization of the objective function

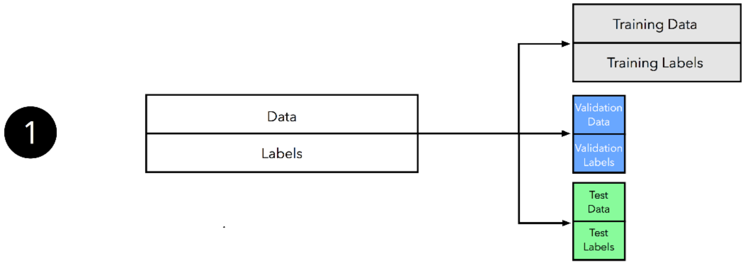
***** with exception of lazy learners

Model selection & optimal hyperparameters: holdout (3-way)

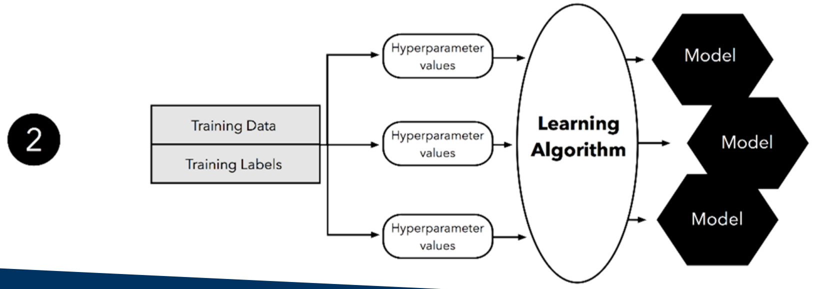


Model selection & optimal hyperparameters: holdout (3-way)

1. The data is split into three subsets (usually random with stratification):
   - training set for model fitting
   - validation set for model selection
   - test set for the final evaluation of the selected model

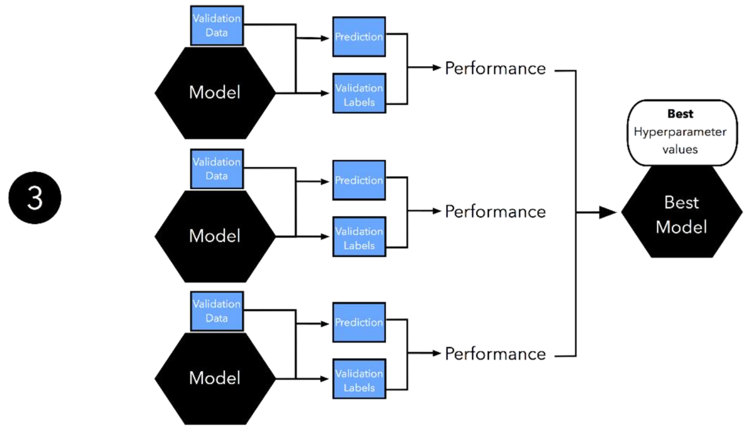


2. Training the model (training set) with **different hyperparameters configurations**
   - For instance: KNN with different values for K, SVM (with different kernels), Neural Networks (with different number of layers/ number of units), and so on

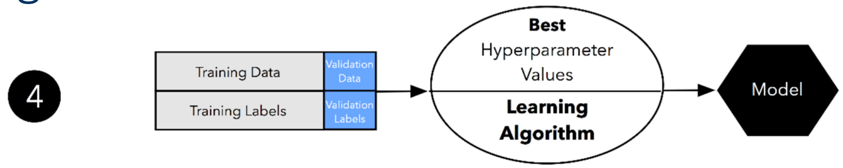


Model selection & optimal hyperparameters: holdout (3-way)

3. Evaluate the performance of each model (for each hyperparameter configuration) on the validation set and choose the hyperparameter conf. associated with the best performance -> Model selection stage

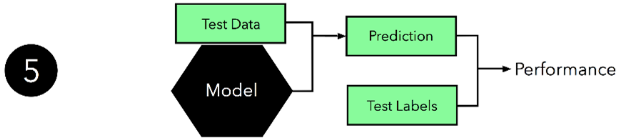


4. After model selection, merge the training and validation set and use the best hyperparameter conf. from the previous step to fit a model to this larger dataset

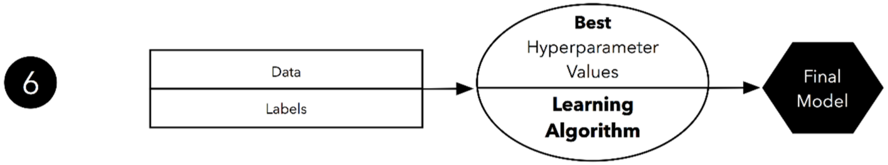


Model selection & optimal hyperparameters: holdout (3-way)

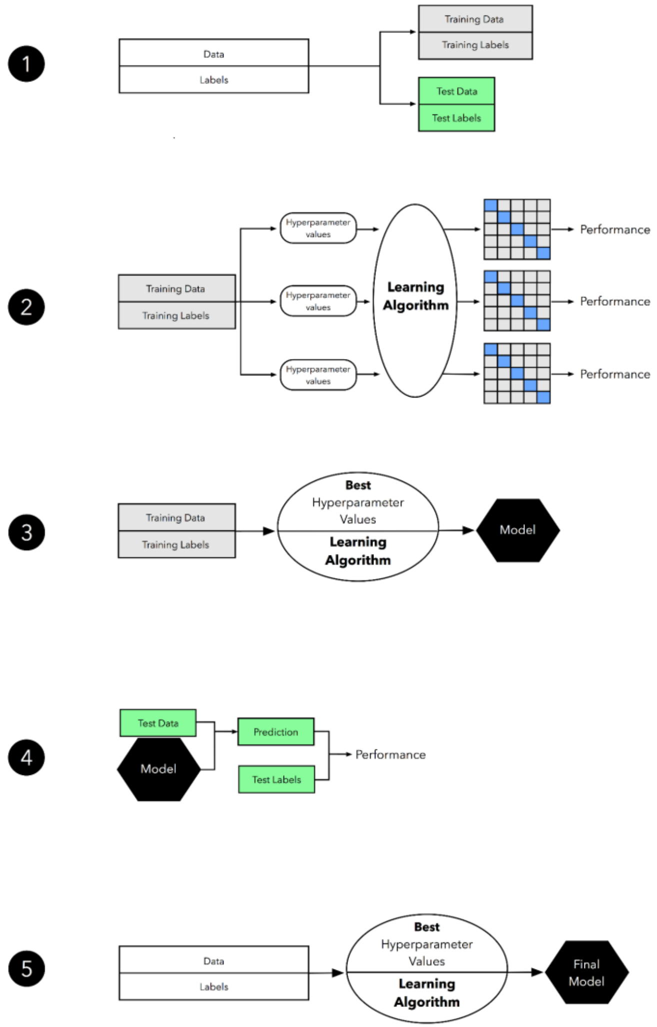
5. Use the independent test set to estimate the generalization performance of the model obtained in step 4



6. Merge training and test set and fit a model (best hyperparameter conf.) to all data points for model deployment

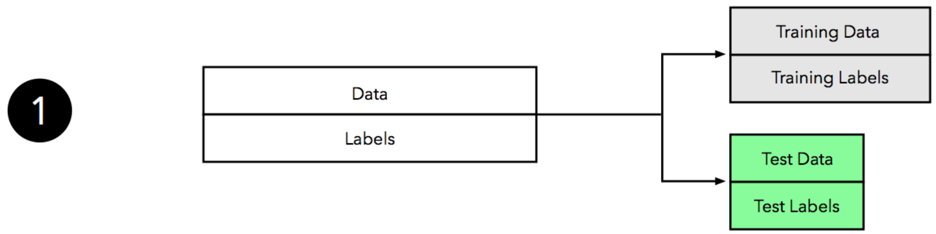


Model selection & optimal hyperparameters: K-Fold Cross Validation



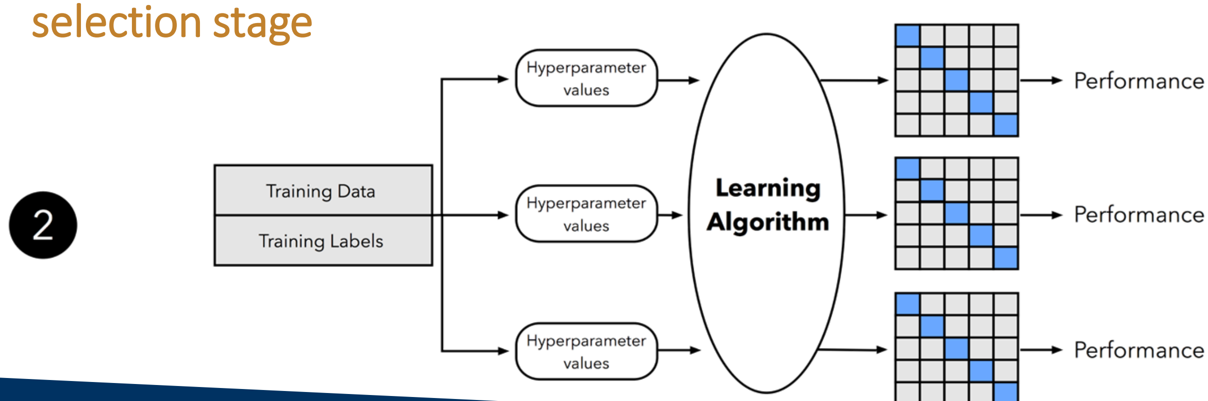
Model selection & optimal hyperparameters: K-Fold Cross Validation

1. The data is split into two subsets (usually random with stratification):
   - training set for model fitting
   - test set for the final evaluation of the selected model



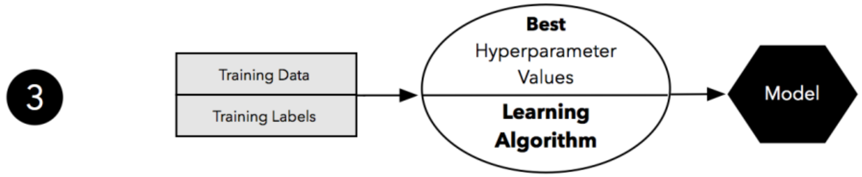
2. Apply k-fold cross-validation on the training set, for each hyperparameter configuration (total number of training run =: *k* x #hyperparameter conf.)

- choose the hyperparameter conf. associated with the best performance -> Model selection stage

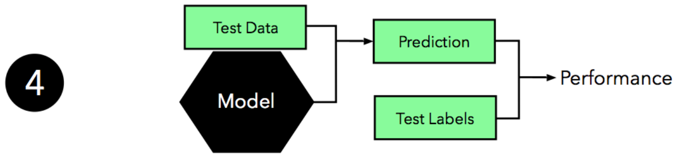


Model selection & optimal hyperparameters: K-Fold Cross Validation

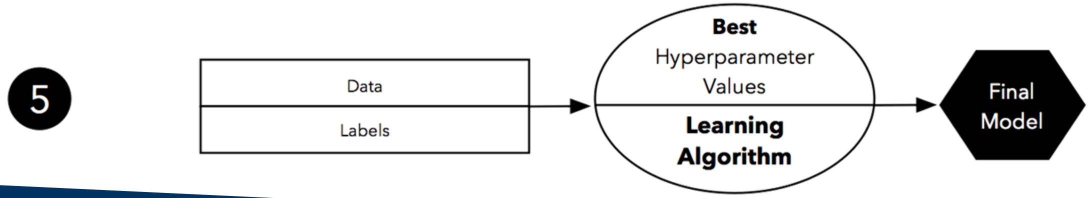
3. With the full training set: fit the model with the hyperparameter conf. that correspond to the best-performing model



4. Use the independent test set to estimate the generalization performance of the model obtained in step 3 (**evaluate on the held-out test set**)



5. Merge training and test set and fit a model (best hyperparameter conf.) to all data points for model deployment



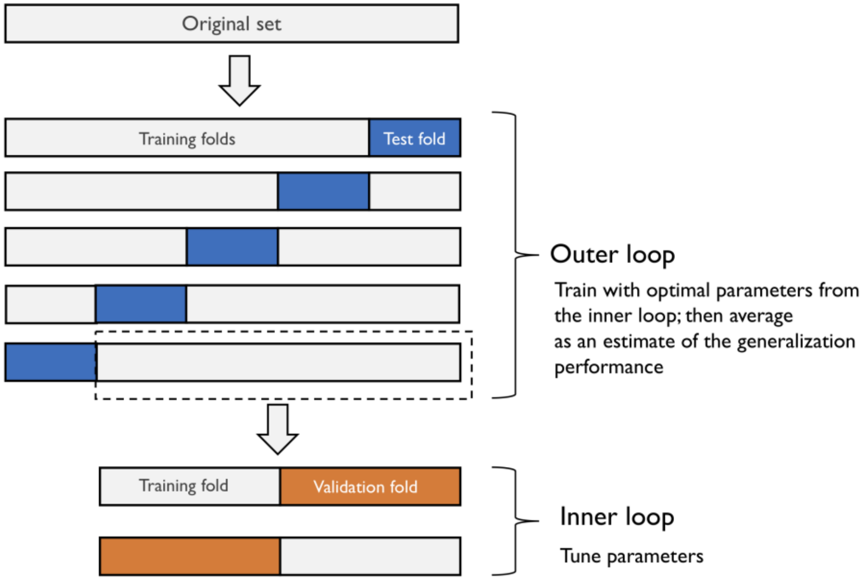
Model selection & optimal hyperparameters: Nested Cross-Validation

Small data sets

- reserving data for independent **test sets is not feasible**

Nesting of two K-Fold Cross-Validation loops:

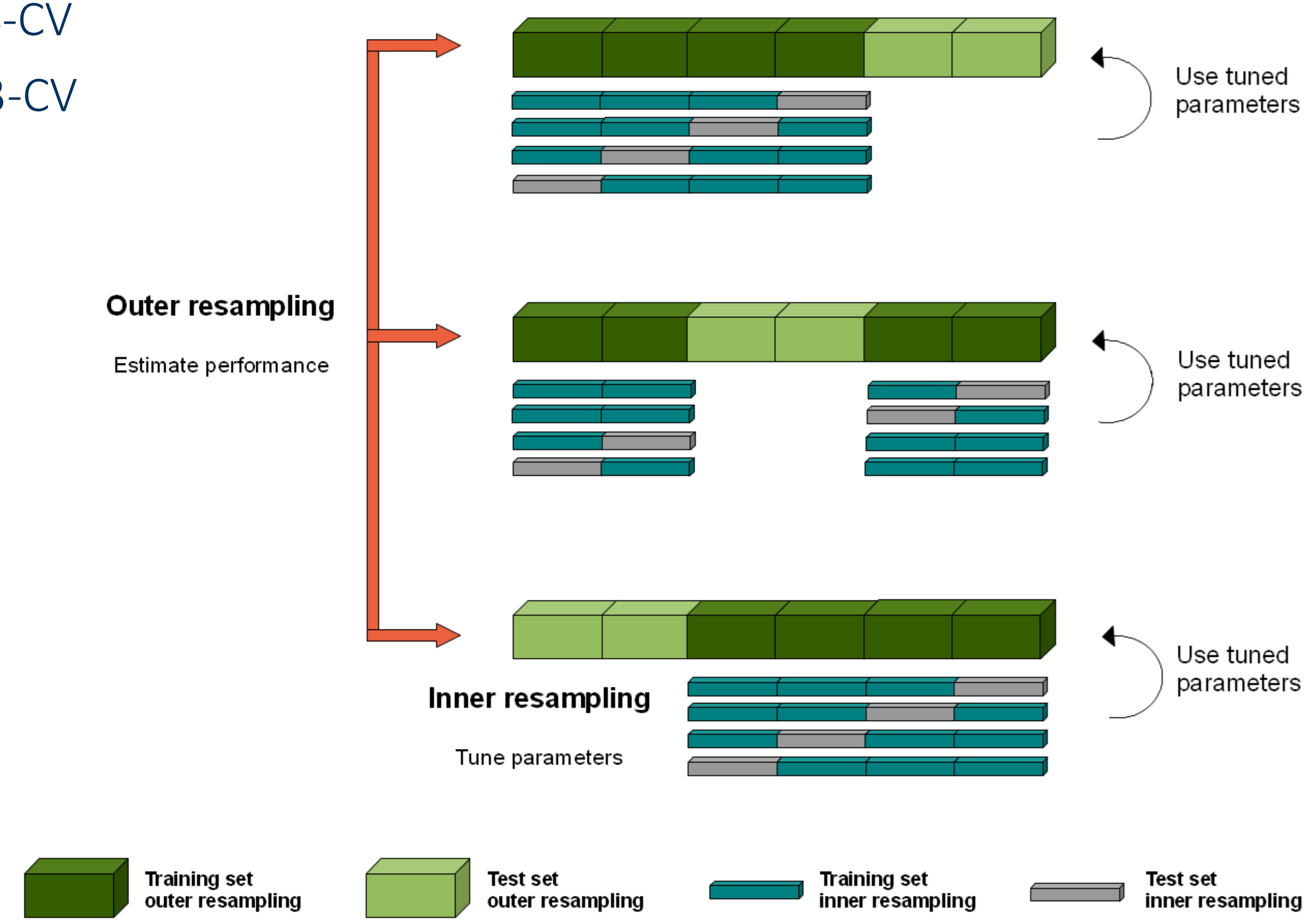
- inner loop: for the model selection
- outer loop: estimating the generalization performance



Has recently emerged as one of the popular or somewhat recommended methods for comparing machine learning algorithms (Iizuka et al., 2003, Varma and Simon, 2006)

Model selection & optimal hyperparameters: Nested Cross-Validation

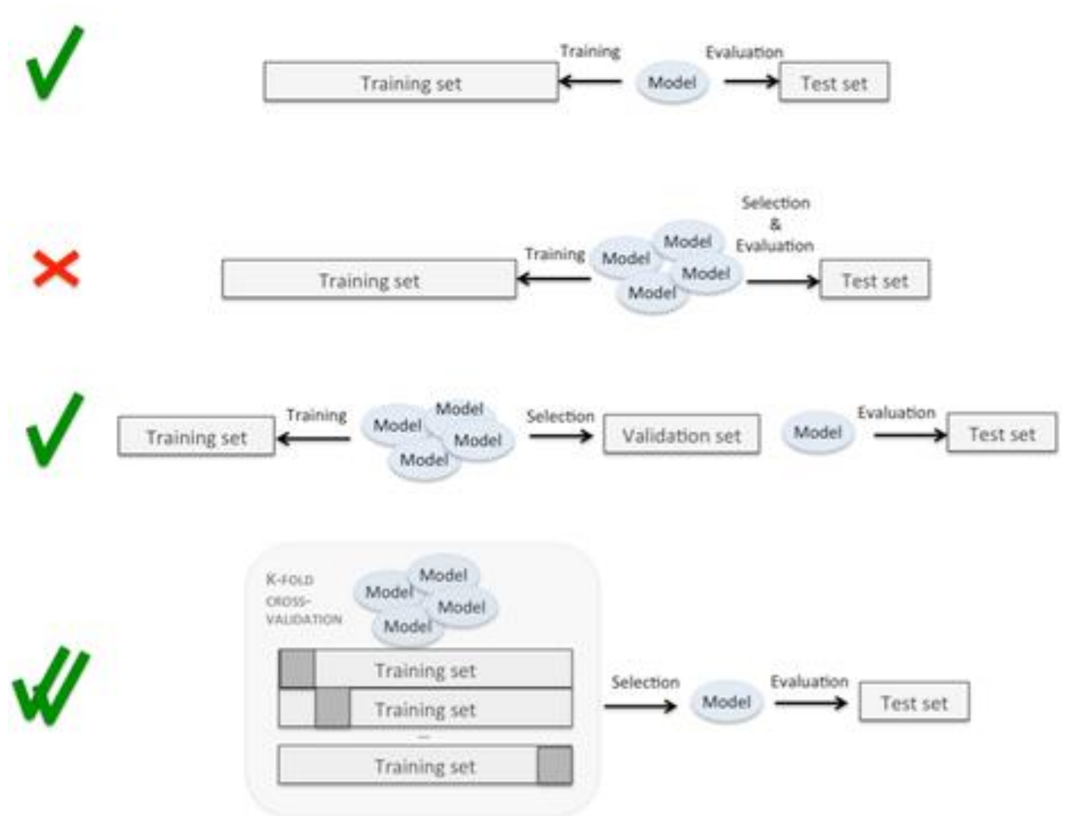
- inner loop: 4-CV
- outer loop: 3-CV



https://mlr.mlr-org.com/articles/tutorial/nested_resampling.html#feature-selection-1

How to evaluate a model?

- Just train a simple model
- Train a model and optimize (tune) its hyperparameters



Contents

Comparison of models

- models were trained and evaluated on the same training and testing sets
- any of the Model evaluation techniques can be used (but the same for all models under comparison)

Comparing the performance of two models:

- Paired t-test
- Mcnemar’s test

Comparing the performance of two or more models:

- F-test

Comparison of 2 models: paired t-test

Use the paired t-test with the goal of comparing the average performance of both classifiers:

$$H_0\colon \mu_1 = \mu_2 \quad \text{vs} \quad H_1\colon \mu_1 \neq \mu_2$$

- H_0: there is no difference among two models, i.e. the true difference is 0 and any differences in performance are attributed to chance

- H_0 is rejected if the result of the paired t-test has a **p-value < α** (α significance level):
  - If **p-value < α** , then H_0 is rejected with **(1 −α) confidence**

- **p-value** is the probability of observing a difference as large as the sample difference given H_0

Comparison of 2 models: paired t-test

Consider that models 1 and 2 were trained with a 10-fold CV technique

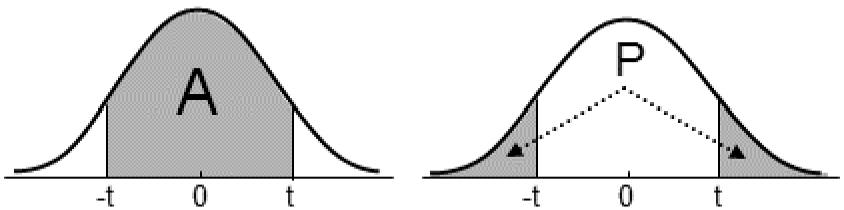
Are the models' performance different?

- accuracy estimates (or error) of classifiers A and B on fold $\boldsymbol{k}$: $p_k^{(m1)}$ and $p_k^{(m2)}$
- compute the paired difference for each fold $\boldsymbol{k}$: $p_k = p_k^{(m1)} - p_k^{(m2)}$
- H_0 is whether p_k has mean $\mathbf{0}$

$$m = \frac{\sum_k p_k}{k} \qquad\qquad s^2 = \frac{\sum_k (p_k - m)^2}{k-1}$$

- the **t score** is estimated as $\boldsymbol{t} = \sqrt{k}\,\frac{m}{s}$
- if $t \in \left]-t_{\alpha/2,(k-1)}, t_{\alpha/2,(k-1)}\right[$, then H_0 is not rejected with **(1 −α) confidence:**
  - **Reject the null hypothesis that the two models' performances are equal**

Comparison of 2 models: paired t-test



- Area A ≡ (1-α) -> confidence level
- Area P ≡ α -> significance level
- (k-1) degrees of freedom (DF)

DF \ A / P	0.90 / 0.10	0.95 / 0.05	0.99 / 0.01	0.995 / 0.005	0.998 / 0.002
1	6.314	12.706	63.657	127.321	318.309
2	2.920	4.303	9.925	14.089	22.327
3	2.353	3.182	5.841	7.453	10.215
4	2.132	2.776	4.604	5.598	7.173
5	2.015	2.571	4.032	4.773	5.893
6	1.943	2.447	3.707	4.317	5.208
7	1.895	2.365	3.499	4.029	4.785
8	1.860	2.306	3.355	3.833	4.501
9	1.833	2.262	3.250	3.690	4.297
10	1.812	2.228	3.169	3.581	4.144
…..	….	….	….	…	….

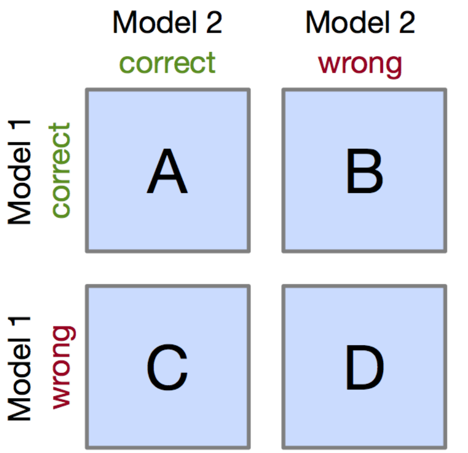
Example: Difference of two models with mean *m = 0.05* and standard deviation *s = 0.002,* in *10* folds

- **t score:** $\boldsymbol{t} = \sqrt{10}\,\frac{0.05}{0.002} = 79.05$
- t-value: $t_{\frac{\alpha}{2},9} = 2.26$ (α=5%)
- $\boldsymbol{t} = 79.05 \notin \,]-2.26, 2.26[$

The nul hypothese H_0 is **rejected**

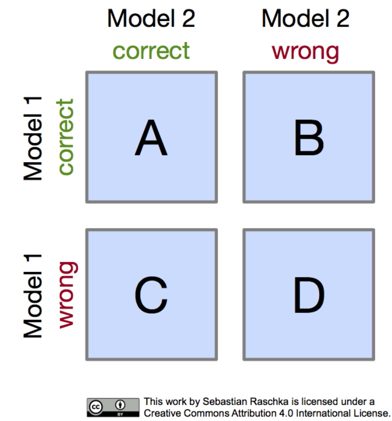
Comparison of 2 models: McNemar’s test

- **more robust alternative**
- also referred to as "within-subjects chi-squared test”
- applied to paired nominal data based on a contingency table
- compares the predictions of two models to each other



Comparison of 2 models: McNemar's test

Compute accuracies of models 1 and 2



- Accuracy_m1 = (A+B) / n
- Accuracy_m2 = (A+C) / n
  - where n = (A+B+C+D) is the total number of test exemples
- Cells B and C (the off-diagonal entries), indicate how the models differ

- H_0 is whether $Prob(B)$ and $Prob(C)$ are the same
- compute the **p-value**
- if **p-value < α** , then H_0 is rejected with **(1 –α) confidence:**
  - **Reject the null hypothesis that the two models' performances are equal**

Comparison of models: F-test for comparing multiple models

Use the F test with the goal of comparing the performance of multiple models:

$$H_0\colon p_1 = p_2 = \cdots = p_L$$

H_0: there is no difference among the performance of the models, i.e. the ***L*** models don't perform differently

Performance of the classifiers evaluated in the same test set

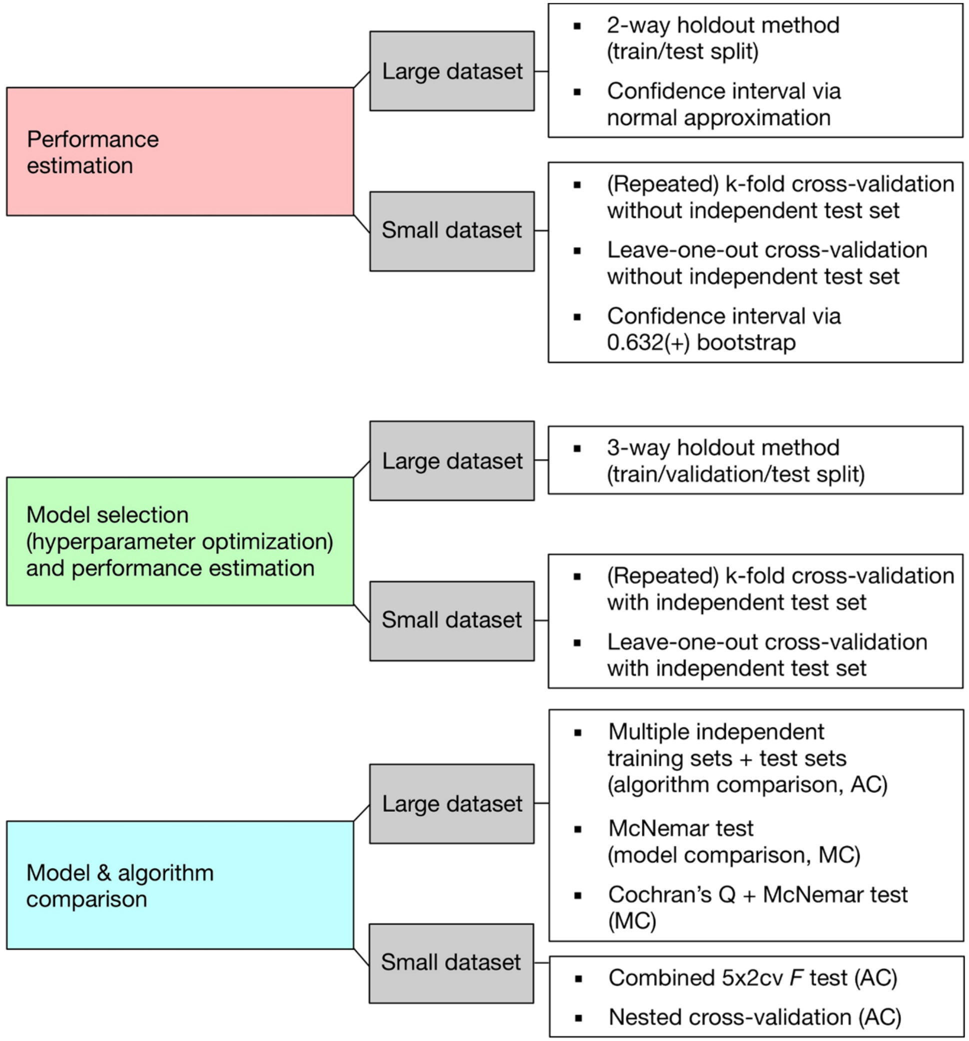
- H_0 is rejected if the result of the **F-test** has a **p-value < α** (α significance level):
  - If **p-value < α** , then H_0 is rejected with **(1 −α) confidence**
    - **There is a difference between the models' performances**
      - perform multiple post hoc pair-wise tests (e.g. McNemar tests with a Bonferroni correction)
        - determine which **pairs of models have different performances**

Contents

Summary

- **Evaluation methodologies**
  - Performance estimation
  - Measures/metrics
    - confusion matrix
    - measures of performance
  - Techniques
    - Holdout
    - K-Fold Cross-Validation
    - Leave-One-Out Cross Validation
- **Model selection & hyperparameter tunning**
  - 3-way Holdout
  - K-Fold Cross-Validation
  - Nested Cross-Validation
- **Comparison of models**

Performance estimation, Model selection & Algorithm selection: recommendations



Bibliography

Introduction to Data Mining, Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, Vipin Kumar, *Pearson*, 2019 (chap 3.5, 3.6)

Data Mining, the Textbook, Charu C. Aggarwal, *Springer*, 2015 (chap 10.9)

Sebastian Raschka, **Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning**. https://arxiv.org/abs/1811.12808