

---

# Exercise Collection – Tuning & Resampling

---

## Contents

<b>Lecture exercises</b>	<b>1</b>
Exercise 1: benchmark experiment – CART vs $k$ -NN . . . . .	1
Exercise 2: benchmark study for classification . . . . .	2
Exercise 3: cross validation for $k$ -NN . . . . .	3
Exercise 4: leave-one-out estimator . . . . .	4
<b>Questions from past exams</b>	<b>5</b>
Exercise 5: WS2020/21, main exam, question 6 . . . . .	5
<b>Ideas &amp; exercises from other sources</b>	<b>5</b>

---

## Lecture exercises

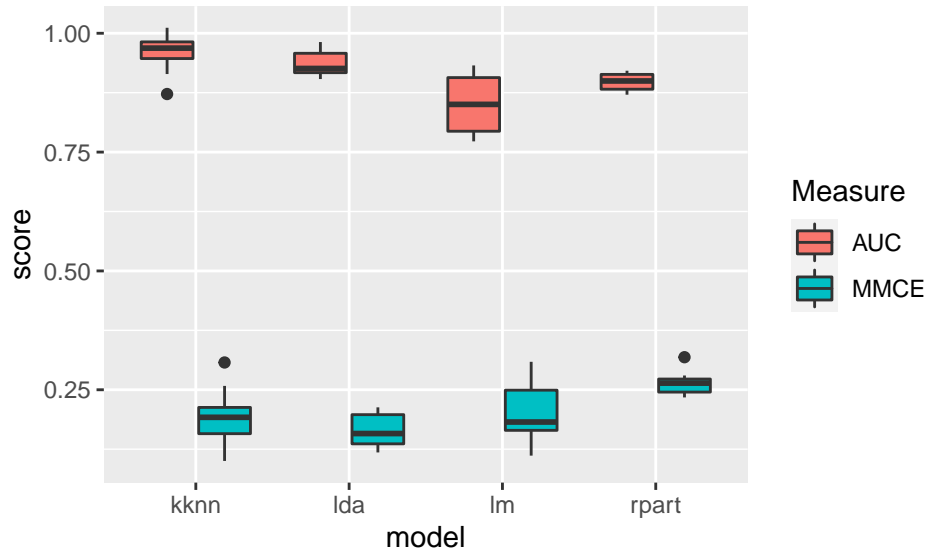
### Exercise 1: benchmark experiment – CART vs $k$ -NN

- (a) Suppose that we want to compare four different models:

<b>Model</b>	<b>Needs Tuning</b>
Logit Model (lm)	No
CART (rpart)	Yes
$k$ -NN (knn)	Yes
LDA (lda)	No

To be able to compare the different models we use a 10-fold cross-validation as outer resampling strategy. Within the tuning of CART and  $k$ -NN we use a 5-fold cross-validation in combination with random search by drawing 200 hyperparameter configurations for each model. Our measure of interest is the AUC.

- (i) To conduct the final benchmark to compare the models, how many models need to be fitted in total?
- (ii) Giving the following benchmark result, which model is best? Explain your decision in one sentence.



- (b) Explain in two sentences what is meant by the *bias - variance trade-off in resampling*.
- (c) Are the following statements true or not, explain your answer in one sentence.
- The bias of the generalization error for 3-fold cross-validation is higher than for 10-fold cross-validation.
  - Every outer loss can also be used as inner loss. Assume any gradient descent based model.

#### Solution 1:

- (a) (i) Important parts:
- Correct number of models for tuning
  - Correctly multiplying tuning models times the two learners that need tuning
  - Correctly adding  $4 \cdot 10$  for learner comparison ()

$$\begin{aligned} \# \text{models} = & \underbrace{4 \cdot 10}_{\# \text{ models outer resampling}} + 2 \cdot \underbrace{10 \cdot \underbrace{5 \cdot 200}_{\# \text{ models for one tuning}}}_{\substack{\# \text{ models for all outer folds for one tuning} \\ \# \text{ models for both tunings}}} = 20040 \end{aligned}$$

- (ii) We would select the k-NN (k-Nearest Neighbors) learner since it achieves the best values for the AUC.
- (b)
- Less data for training leads to higher bias
  - More data for training and less data for evaluation lead to higher variance
- (c) Are the following statements true or not, explain your answer in one sentence.
- True, using 3-fold cross-validation leads to smaller train sets and therefore we are not able to learn as much as for, e.g., 10-fold cross-validation.
  - False, the outer loss doesn't have as many restrictions as the inner loss, e.g. the outer loss doesn't have to be differentiable.

## Exercise 2: benchmark study for classification

We want to conduct a small benchmark study:

- (a) Set up at least two datasets with a discrete target of your choice and save them in a list.
- (b) Set up at least three classifiers of your choice (that you are familiar with) and save them in a second list.
- (c) Create a resample description. Use an adequate method, depending on the size of your datasets and the complexity of the learners, so that the benchmark doesn't need hours to compute.
- (d) Run and visualize your benchmark study.

### Solution 2:

See R code

## Exercise 3: cross validation for $k$ -NN

Use the `bodyfat` dataset from the package `TH.data` for this exercise.

- (a) Create a `mlr` regression task for the `bodyfat` data with `DEXfat` as target variable.
- (b) Fit a K-Nearest-Neighbor regression model
- (c) Visualize the predictions for the features `waistcirc` and `anthro3c` separately as well as together in one plot.
- (d) Resample the `bodyfat` dataset with 10-fold crossvalidation and calculate the mean squared error and the median absolute error.

### Solution 3:

See R code

## Exercise 4: leave-one-out estimator

In this exercise we take a look at the leave-one-out-estimator. We will use a data independent model, so our data are all i.i.d. bernoulli(0.5) distributed labels,  $\{0, 1\}$ ,  $Y_1, \dots, Y_n$ . Our (a bit strange) rule results after looking at the training data always in a constant estimation. It works like that: If the training data consists of an odd number of 1s it will estimate a 1, otherwise a 0.

- (a) What is the expected error of this rule?
- (b) Now we estimate this expected error on the training data with the leave-one-out-estimator. What expected value and variance has the result? How would you interpret this result?

### Solution 4:

- (a) Let  $Y_1, \dots, Y_n, Y$  all be i.i.d and Bernoulli- $(\frac{1}{2})$  distributed.  $Y_1, \dots, Y_n$  is the training data,  $Y$  is a new test observation and  $\hat{Y}$  the prediction of our data independent rule for  $Y$ .

It follows:

$$P(\hat{Y} \neq Y) = P(\hat{Y} = 1)(1 - 0.5) + (1 - P(\hat{Y} = 1))0.5$$

and (this is not really required but will be useful later)

$$P(\hat{Y} = 1) = \frac{1}{2}.$$

This is the case because in the product space for  $Y_1, \dots, Y_n$  all events  $Y_1 = y_1, \dots, Y_n = y_n$  have the same probability ( $0.5^n$ ) and exactly half of the events have an even number of 1s, the other half an odd number.

So it follows that:

$$P(\hat{Y} \neq Y) = 0.5$$

- (b) Let  $\hat{L}$  be the leave-one-out (LOO) estimator of  $Y_1, \dots, Y_n$  and  $\hat{Y}_i$  the prediction of our data independent rule for  $Y_i$  with LOO. This means the estimation is based on  $Y_1, \dots, Y_n$  without  $Y_i$ .

It follows:

$$\hat{L} = \frac{1}{n} \sum_{i=1}^n I(\hat{Y}_i \neq Y_i).$$

$\hat{L}$  can only have the value 1 or 0. Which can be explained like this:

Assume that  $Y_1(\omega), \dots, Y_n(\omega)$  have an even number of 1s. Now drop one  $Y_i(\omega)$ . If it was a 0, the remaining observations still have an even number of 1s, a 0 will be predicted and  $\hat{Y}_j(\omega) = 0 = Y_j(\omega)$ . If it is a 1, it results in an odd number of 1s, a 1 will be predicted and  $\hat{Y}_j(\omega) = 1 = Y_j(\omega)$ . So,  $\hat{L}(\omega) = 0$ .

Now assume that  $Y_1(\omega), \dots, Y_n(\omega)$  have an odd number of 1s. Again, drop one  $Y_i(\omega)$ . If it was a 0, the remaining observations still have an odd number of 1s, a 1 is predicted and  $\hat{Y}_j(\omega) \neq Y_j(\omega)$ . If it is a 1, it results in an even number of 1s, a 0 will be predicted and  $\hat{Y}_j(\omega) \neq Y_j(\omega)$ . So,  $\hat{L}(\omega) = 1$ .

This means that  $\hat{L}$  is a Bernoulli distributed random variable. An even number of 1s in  $Y_1, \dots, Y_n$  has the same probability as an odd number, so  $\hat{L} \sim \text{Bernoulli}(0.5)$  and consequently  $E(\hat{L}) = 1/2$  and  $Var(\hat{L}) = 1/4$ .

The expected value is also correct like in a). But the variance is very high. The LOO can only be calculated once and it behaves like a fair coin toss. It will produce an error of 0% or an error of 100%. The example is rather theoretical but shows the disadvantageous property of a high variance in the LOO estimator.

---

## Questions from past exams

### Exercise 5: WS2020/21, main exam, question 6

Assume a polynomial regression model with a continuous target variable  $y$  and a continuous,  $p$ -dimensional feature vector  $\mathbf{x}$  and polynomials of degree  $d$ , i.e.,

$f(\mathbf{x}^{(i)}) = \sum_{j=1}^p \sum_{k=0}^d \theta_{j,k} (\mathbf{x}_j^{(i)})^k$  and  $y^{(i)} = f(\mathbf{x}^{(i)}) + \epsilon^{(i)}$  where the  $\epsilon^{(i)}$  are iid with  $\text{Var}(\epsilon^{(i)}) = \sigma^2 \forall i \in \{1, \dots, n\}$ .

- (a) On a given data set with sample size  $n = 1000$ ,  $p = 1$  feature and degree  $d = 1$  we want to estimate the generalization error. Describe the advantages and disadvantages of the following three resampling strategies. Additionally, state which strategy you would use here.
- (i) hold-out sampling, i.e., a single train-test split
  - (ii) leave-one-out cross-validation, i.e.,  $n$ -fold cross-validation
  - (iii) 5-fold cross-validation

#### Solution 5:

- (a)
- (i): fastest, but very dependent on the given split, i.e., can vary quite a lot with varying seed
  - (ii): slow, very robust - does not depend on seed  $\Rightarrow$  Perhaps best in that case, because linear model is fast
  - (iii): trade-off: higher bias as loocv but not that dependent on the seed  $\Rightarrow$  Depending on computation time perhaps better than loocv

---

## Ideas & exercises from other sources