

# \*\*\* Applied Machine Learning Fundamentals \*\*\*

## Evaluation of ML Models

Daniel Wehner, M.Sc.

SAP SE / DHBW Mannheim

Winter term 2020/2021



Find all slides on [GitHub](#) (DaWe1992/Applied\_ML\_Fundamentals)

# Lecture Overview

Unit I	Machine Learning Introduction
Unit II	Mathematical Foundations
Unit III	Bayesian Decision Theory
Unit IV	Probability Density Estimation
Unit V	Regression
Unit VI	Classification I
<b>Unit VII</b>	<b>Evaluation</b>
Unit VIII	Classification II
Unit IX	Clustering
Unit X	Dimensionality Reduction

# Agenda for this Unit

## ① Evaluation Methods and Data Splits

Introduction

Cross-Validation / LOO-Validation

Data Splits

## ② Evaluation Metrics

Confusion Matrices

Drawback of Accuracy

Precision, Recall and F1-Score

ROC and AUC

## ③ Cost-sensitive Evaluation

Misclassification Costs

Expected Costs and Cost Ratio

Selection of optimal Classifiers

Calibration of Thresholds

## ④ Miscellaneous

Evaluation of Regressors

Grid Search and Random Search

Bias and Variance

## ⑤ Wrap-Up

Summary

Self-Test Questions

Lecture Outlook

Recommended Literature and further Reading

Meme of the Day

Section:  
**Evaluation Methods and Data Splits**



# Evaluation of trained Models

- ① **Validation through experts:** A domain expert checks plausibility
  - Subjective, time-intensive, costly
  - Often the only option
- ② **Validation on data:** Evaluate performance on a **separate (!)** test set
  - Labeled data is scarce, could be better used for training
  - Fast and simple, no domain knowledge needed
- ③ **On-line validation:** Test model in a fielded application
  - Bad models may be costly
  - Gives the best estimate for the overall utility



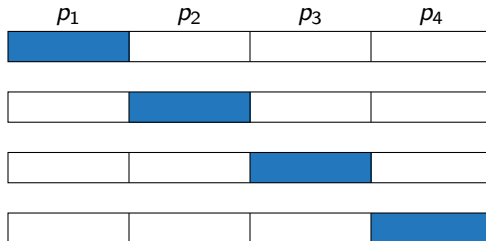
# Out-of-Sample Testing

- The performance cannot be measured on the training data ( $\Rightarrow$  overfitting!)
- Usually, a portion of the available data is reserved for testing
  - $2/3$  for training,  $1/3$  for testing (evaluation)
  - The model is trained on the training set and evaluated on the test set
- **Problems:**
  - Waste of data
  - Labeling may be expensive
- **Solution:** **Cross-Validation (X-Val)**



# Cross-Validation (X-Val)

- Split the data set into  $k$  equally sized partitions  $P = \{p_1, p_2, \dots, p_k\}$
- For each partition  $p_i$  do: use  $p_i$  for testing and  $P \setminus \{p_i\}$  for training
- Average the results; e. g. 4-fold X-Val:



# Leave-One-Out Cross-Validation (LOO X-Val)

- $n$ -fold X-Val
  - $n$  is the number of examples
  - Use  $n - 1$  examples for training, one example for testing
- Properties
  - Makes best use of the data
  - Very expensive for large data sets (large  $n$ )

If  $k$ -fold X-Val is performed, we get  $k$  trained models!

- Which model is used in production?
- **Answer:** None. X-Val is only used for error estimation. The final model is trained on the entire data set





# Three Splits: Train, Dev/Validation, Test

In practice it is common to split the data into three portions:

**Stratified splits** have the same class dist. as the entire data set

- ① **Training set** (used for training as before)
- ② **Dev/Validation set**
  - Used for hyper-parameter tuning of the model
  - Using the test set for that would be cheating
- ③ **Test set**
  - The final model is tested on the test set
  - Test set is used to estimate the **generalization error**

## Section: Evaluation Metrics



# Types of Errors

- **Type I Error:** False negatives
  - An instance which is labeled  $\ominus$  is classified as  $\oplus$
  - E. g. a spam e-mail is not detected
- **Type II Error:** False positives
  - An instance which is labeled  $\oplus$  is classified as  $\ominus$
  - E. g. a non-spam (ham) e-mail is classified as spam

a. k. a.  $\alpha/\beta$  error

Depending on the context the costs of false negatives and false positives can be different!



# Confusion Matrices (two Classes)

- How often is class  $\mathcal{C}_i$  confused with class  $\mathcal{C}_j$ ?
- Calculate **accuracy**:

	Classified $\oplus$	Classified $\ominus$
Is $\oplus$	true positives ( $tp$ )	false negatives ( $fn$ )
Is $\ominus$	false positives ( $fp$ )	true negatives ( $tn$ )

$$accuracy = \frac{tp + tn}{tp + tn + fp + fn}$$

$$error = 1 - accuracy$$

# Confusion Matrices (multiple Classes)

	A	B	C	D	$\Sigma$
A	$n_{A,A}$	$n_{B,A}$	$n_{C,A}$	$n_{D,A}$	$n_A$
B	$n_{A,B}$	$n_{B,B}$	$n_{C,B}$	$n_{D,B}$	$n_B$
C	$n_{A,C}$	$n_{B,C}$	$n_{C,C}$	$n_{D,C}$	$n_C$
D	$n_{A,D}$	$n_{B,D}$	$n_{C,D}$	$n_{D,D}$	$n_D$
$\Sigma$	$\overline{n_A}$	$\overline{n_B}$	$\overline{n_C}$	$\overline{n_D}$	$n$

$$accuracy = \frac{n_{A,A} + n_{B,B} + n_{C,C} + n_{D,D}}{n}$$

# Drawback of Accuracy

- Real-world data sets are usually **imbalanced**, i. e. some classes appear more frequently than others
- **Example:**
  - A data set  $\mathcal{D}$  contains two classes  $\mathcal{C}_1$  and  $\mathcal{C}_2$
  - $\mathcal{C}_1$  appears 99 % of the time,  $\mathcal{C}_2$  1 % of the time
  - It is easy to reach 99 % accuracy by always predicting the majority class
  - **Is this useful?** *Probably not...*

**We need some more sophisticated evaluation metrics!**

# Precision and Recall

**Precision**: Ratio of  $tp$  to all instances predicted as  $\oplus$

$$Precision (P) = \frac{tp}{tp + fp} \quad (1)$$

**Recall** (Sensitivity): Ratio of  $tp$  to all instances actually labeled as  $\oplus$

$$Recall (R) = \frac{tp}{tp + fn} \quad (2)$$

# Precision-Recall-Trade-Off

There is a trade-off between precision and recall:

It is very easy to get 100 % precision:

- Simply classify one instance as  $\oplus$  where you are absolutely sure
- But recall is bad... (*many  $\oplus$ -instances are not detected*)

---

It is also quite easy to achieve 100 % recall:

- Classify all instances as  $\oplus$
- But precision is bad... (*many  $\ominus$ -instances are detected*)

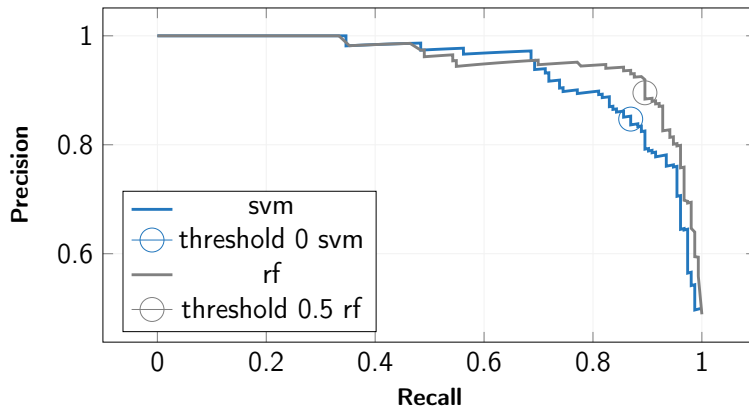


# Precision-Recall Curves / P-R-Curves

- Visualization of the Precision-Recall-trade-off
- Influence precision and recall by changing thresholds
- **Example:**
  - Consider a ranker, e. g. a logistic regression classifier
  - It outputs probabilities for each class
  - The threshold when to predict  $\oplus$  can be changed
  - This has an influence on precision and recall

A P-R-curve plots precision and recall for all possible thresholds.

## Precision-Recall Curves / P-R-Curves (Ctd.)



# Combining Precision and Recall: F1-Score

- When to use precision, when recall?
- This depends on the cost of  $fp$  and  $fn$ 
  - If  $fp$  are expensive  $\Rightarrow$  **use precision!**
  - If  $fn$  are expensive  $\Rightarrow$  **use recall!**
- **F1-score** (*harmonic mean of precision and recall*)

Why the harmonic mean?

$$F_1 = \frac{2 \cdot P \cdot R}{P + R} \quad F_\beta = (1 + \beta^2) \cdot \frac{P \cdot R}{(\beta^2 \cdot P) + R} \quad (\beta \in \mathbb{R}^+) \quad (3)$$

- Large  $\beta$  emphasizes recall

## Calculation for multiple Classes (Example Precision)

- Precision must be calculated for each class separately
- For  $|\mathcal{C}|$  classes we get  $|\mathcal{C}|$  results. **How to combine?**
  - **Macro average:** Calculate  $P$  for each class and average the result

$$P_{macro} = \frac{P_A + P_B + P_C + P_D}{|\mathcal{C}|} \quad (4)$$

- **Micro average:** Sum all  $tp$  and  $fp$  for all classes and calculate  $P$

$$P_{micro} = \frac{tp_A + tp_B + tp_C + tp_D}{(tp_A + tp_B + tp_C + tp_D) + (fp_A + fp_B + fp_C + fp_D)} \quad (5)$$

# Calculation for multiple Classes (Example Precision)

	A	B	C	D	$\Sigma$
A	40	12	4	8	64
B	7	51	2	0	60
C	2	17	27	11	57
D	39	4	15	8	66
$\Sigma$	88	84	48	27	247

Cols: Prediction  
 Rows: Gold label

$$P_A = \frac{40}{40 + 48} = 0.45$$

$$P_B = 0.61$$

$$P_C = 0.56$$

$$P_D = 0.30$$

$$P_{macro} = \frac{0.45 + 0.61 + 0.56 + 0.30}{4} = 0.48$$

$$P_{micro} = \frac{40 + \dots + 8}{(40 + \dots + 8) + (48 + \dots + 19)} = 0.51$$

# ROC-Curves

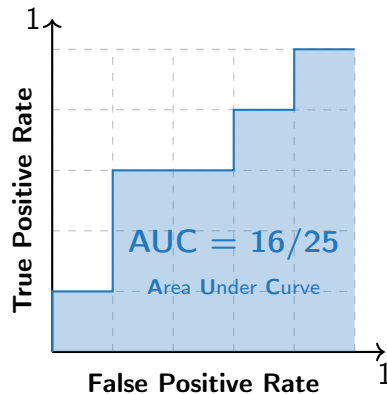
- ROC = Receiver Operating Characteristic
- Borrowed from signal theory (*hence the name*)
- Uses *true positive rate* (recall) and *false positive rate* =  $\frac{fp}{fp+tn}$

## General procedure:

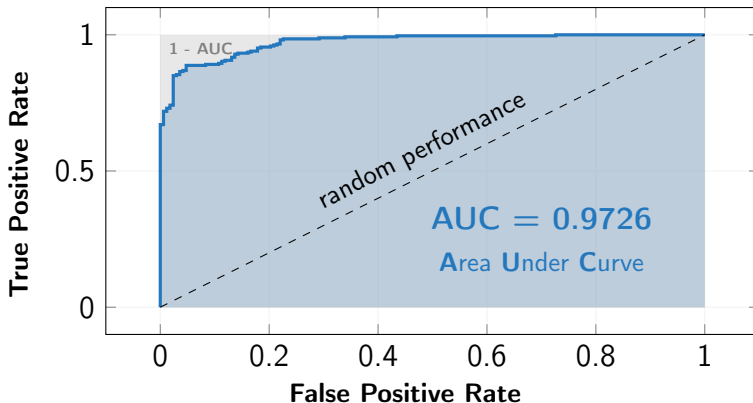
- Rank test instances by decreasing certainty of class  $\oplus$
- Start at the origin (0, 0)
- If the next instance in the ranking is  $\oplus$ : move  $1/|\oplus|$  up
- If the next instance in the ranking is  $\ominus$ : move  $1/|\ominus|$  right

# Sample ROC-Curve I

Rank	Prob.	True class
1	0.95	$\oplus$
2	0.85	$\ominus$
3	0.78	$\oplus$
4	0.75	$\oplus$
5	0.62	$\ominus$
6	0.41	$\ominus$
7	0.37	$\oplus$
8	0.22	$\ominus$
9	0.15	$\oplus$
10	0.05	$\ominus$



## Sample ROC-Curve II





# ROC-Curve Interpretation

- AUC can be interpreted as the probability of a positive example always being listed before a negative example
- A high AUC value entails a good class separation:
  - AUC = 1.0:** All  $\oplus$  listed before all  $\ominus$  (desiderata)
  - AUC = 0.5:** Random ordering
  - AUC = 0.0:** All  $\ominus$  listed before all  $\oplus$  (not the worst case  $\Rightarrow$  Invert classification)

**Analogy:** It is like a quiz. But you can answer those questions first where you feel the most certain (ranking). If you answer the first questions wrong, you don't perform well  $\Rightarrow$  **small AUC**.

Section:  
**Cost-sensitive Evaluation**



# Cost-Sensitive Evaluation

- Predicting class  $\mathcal{C}_i$  instead of the correct class  $\mathcal{C}_j$  is associated with a cost-factor  $c(\mathcal{C}_i|\mathcal{C}_j)$
- Usually, there are only costs for wrong predictions
- 0/1-Loss:

$$c(\mathcal{C}_i|\mathcal{C}_j) = \begin{cases} 0 & \text{if } i = j \\ 1 & \text{if } i \neq j \end{cases}$$

- General case (two class problems):

	Classified $\oplus$	Classified $\ominus$
Is $\oplus$	$c(\oplus \oplus)$	$c(\ominus \oplus)$
Is $\ominus$	$c(\oplus \ominus)$	$c(\ominus \ominus)$

# Cost-Sensitive Evaluation Examples

- **Loan applications**

Rejecting applicants who will not pay back

→ **no costs**

Accepting applicants who will pay back

→ **gain**

Accepting applicants who will not pay back

→ **big loss**

Rejecting applicants who would pay back

→ **loss**

- **Spam-mail filtering**

- **Medical diagnosis**

- ...

## Expected Costs / Loss and Cost Ratio

- Expected loss  $\mathcal{L}$ :

$$\mathcal{L} = tpr \cdot c(\oplus|\oplus) + fpr \cdot c(\oplus|\ominus) + fnr \cdot c(\ominus|\oplus) + tnr \cdot c(\ominus|\ominus) \quad (6)$$

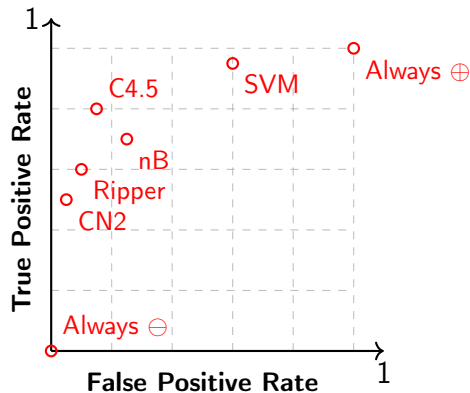
- If there are no costs for a correct classification:

$$\mathcal{L} = fpr \cdot c(\oplus|\ominus) + fnr \cdot c(\ominus|\oplus) \quad (7)$$

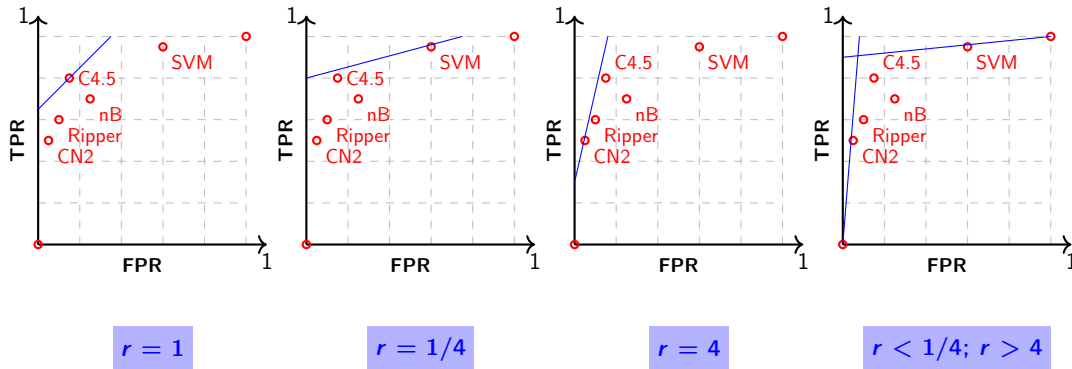
- Cost ratio** (*false positives are  $r$  times as expensive as false negatives*)

$$r = \frac{c(\oplus|\ominus)}{c(\ominus|\oplus)} = \frac{c_{fp}}{c_{fn}} \quad (8)$$

# Classifiers in ROC-Space – Example

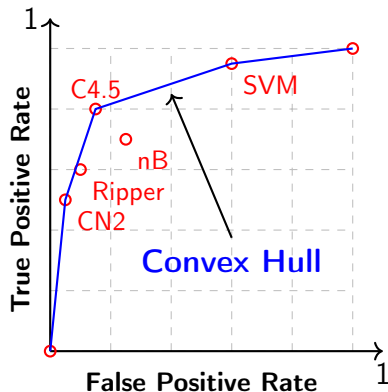


## Classifiers in ROC-Space – Example (Ctd.)



## Classifiers in ROC-Space – Example (Ctd.)

Classifiers on the convex hull minimize costs for some cost ratio.  
Classifiers below the convex hull are always suboptimal.





## Classifiers in ROC-Space (Ctd.)

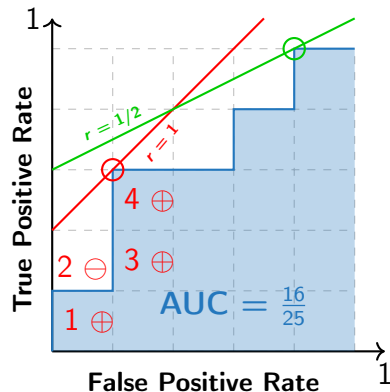
- It is possible to reach any point on the convex hull
- **Interpolation of two adjacent classifiers in ROC-space:**
  - Classifier 1:  $tpr_1$  and  $fpr_1$
  - Classifier 2:  $tpr_2$  and  $fpr_2$
  - If classifier 1 is used to predict  $q \cdot 100\%$  and classifier 2 for the rest:

$$tpr_{inter} = q \cdot tpr_1 + (1 - q) \cdot tpr_2$$

$$fpr_{inter} = q \cdot fpr_1 + (1 - q) \cdot fpr_2$$

# Calibrating Thresholds

Rank	Prob.	True class
1	0.95	$\oplus$
2	0.85	$\ominus$
3	0.78	$\oplus$
4	0.75	$\oplus$
5	0.62	$\ominus$
6	0.41	$\ominus$
7	0.37	$\oplus$
8	0.22	$\ominus$
9	0.15	$\oplus$
10	0.05	$\ominus$



Section:  
**Miscellaneous**



# Evaluation of Regressors

- Coefficient of determination  $R^2$ :

$$R^2 = \frac{\sum_{i=1}^n (h_{\theta}(\mathbf{x}^{(i)}) - \bar{y})^2}{\sum_{i=1}^n (y^{(i)} - \bar{y})^2} = \frac{\text{Variance explained by model}}{\text{Total variance}} \quad R^2 \in [0, 1] \quad (9)$$

- Root mean square error (RMSE):

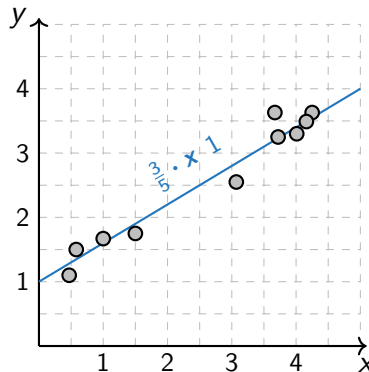
$$RMSE = \left( \frac{1}{n} \cdot \sum_{i=1}^n (h_{\theta}(\mathbf{x}^{(i)}) - y^{(i)})^2 \right)^{1/2} \quad (10)$$

- Mean absolute error (MAE):

$$MAE = \frac{1}{n} \cdot \sum_{i=1}^n |h_{\theta}(\mathbf{x}^{(i)}) - y^{(i)}| \quad (11)$$

# Evaluation of Regressors (Ctd.)

$x^{(i)}$	$y^{(i)}$	$h_{\theta}(x^{(i)})$
0.47	1.10	1.28
0.58	1.50	1.35
1.00	1.67	1.60
1.50	1.75	1.90
3.07	2.55	2.84
3.67	3.63	3.20
3.72	3.25	3.23
4.01	3.30	3.41
4.16	3.49	3.50
4.25	3.63	3.55
$\bar{y} = 2.59$		



# Evaluation of Regressors (Ctd.)

- Coefficient of determination:

$$R^2 = \frac{(1.28 - 2.59)^2 + \dots + (3.55 - 2.59)^2}{(1.10 - 2.59)^2 + \dots + (3.63 - 2.59)^2} = \frac{7.97}{8.89} = 0.90 \quad (12)$$

- Root mean square error:

$$RMSE = \left( \frac{1}{10} \cdot [(1.28 - 1.10)^2 + \dots + (3.55 - 3.63)^2] \right)^{1/2} = 0.19 \quad (13)$$

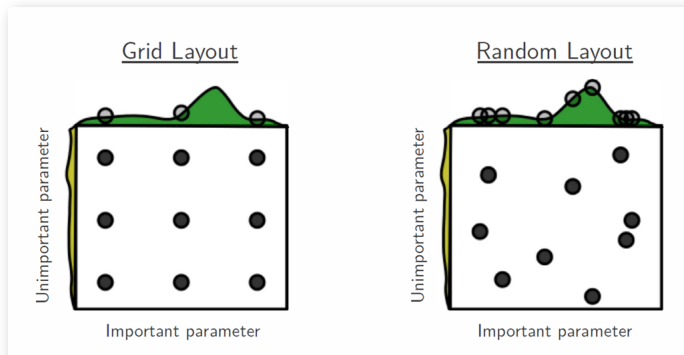
- Mean absolute error:

$$MAE = \frac{1}{10} \cdot (|1.28 - 1.10| + \dots + |3.55 - 3.63|) = 0.15 \quad (14)$$

# Grid Search

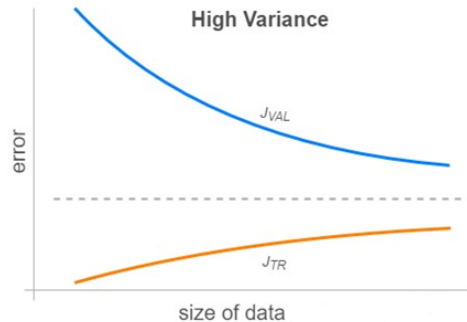
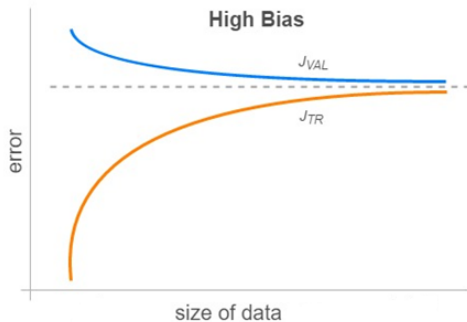
- **Grid search** is applied to find **optimal parameter settings**
- For the optimization the **dev** data set is used
- We have to specify the search space / ranges of parameter values
- Grid search will try **all parameter combinations** to find the best model
  - Computationally very expensive
  - Scikit-learn provides parameters to parallelize the search  
(`n_jobs=-1`  $\Rightarrow$  use all cores available)
  - May not find the optimal setting  $\Rightarrow$  **random search**

# Grid Search vs. random Search





# Bias and Variance

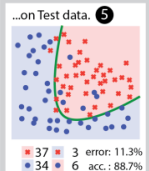


Use early stopping!

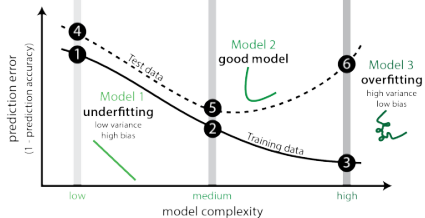
Model 1...



Model 2...



Model 3...



Section:  
**Wrap-Up**



# Summary

- **Out-of-sample testing:** Split data into train, dev and test sets
- Cross-validation makes **maximum use of the data**
- Confusion matrices reveal **which classes are frequently confused**
- Precision, recall and F1 are **more robust w. r. t. imbalanced data sets**
- ROC curves are used for the evaluation of rankers
- Different classifiers can be optimal assuming a different cost model
- Hyper-parameters are optimized using **grid search** or **random search**
- Keep the **bias-variance trade-off** in mind!



# Self-Test Questions

- 1 Why should you split the data into train, dev and test sets?
- 2 You perform 10-fold cross validation. How many models do you have to learn? Which one do you use in production?
- 3 What is the problem with accuracy?
- 4 Why do we apply the harmonic mean to compute the F1 score?
- 5 Your model gets an AUC value of 0. What does this mean?
- 6 Random search is usually preferred to optimize hyper-parameters. Why?
- 7 Your model does not perform well due to its high bias. Your boss suggests adding more training data. How would you respond?

# What's next...?

Unit I	Machine Learning Introduction
Unit II	Mathematical Foundations
Unit III	Bayesian Decision Theory
Unit IV	Probability Density Estimation
Unit V	Regression
Unit VI	Classification I
Unit VII	Evaluation
<b>Unit VIII</b>	<b>Classification II</b>
Unit IX	Clustering
Unit X	Dimensionality Reduction

# Recommended Literature and further Reading I

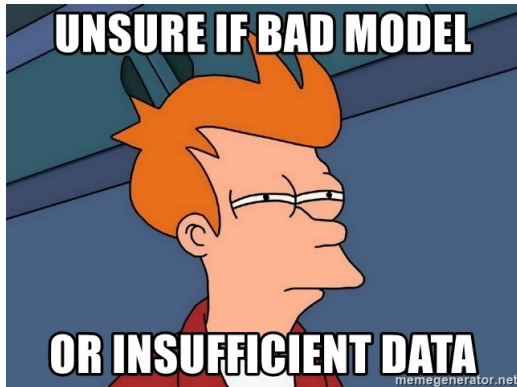


## [1] Machine Learning

*Tom Mitchell. McGraw-Hill Science. 1997.*

→ [Link](#), cf. chapter 5

## Meme of the Day





Thank you very much for the attention!

**Topic:** \*\*\* Applied Machine Learning Fundamentals \*\*\* Evaluation of ML Models

**Term:** Winter term 2020/2021

**Contact:**

Daniel Wehner, M.Sc.

SAP SE / DHBW Mannheim

[daniel.wehner@sap.com](mailto:daniel.wehner@sap.com)

Do you have any questions?