UNIVERSITY
OF OSLO

IN4310/3310 Performance estimation and generalisation beyond the development dataset

---

Slides by Dhanajay Tomar
(modified by Vilde Schulerud Bøe)



EVALUATION

Outline

1. Internal testing (recap)
2. External testing
3. Facilitating generalisation
4. Some performance metrics
5. Estimator uncertainty

Outline

1. Internal testing (recap)

2. External testing

3. Facilitating generalisation

4. Some performance metrics

5. Estimator Uncertainty

Train and test subsets

- Randomly partition development dataset into two disjoint subsets:



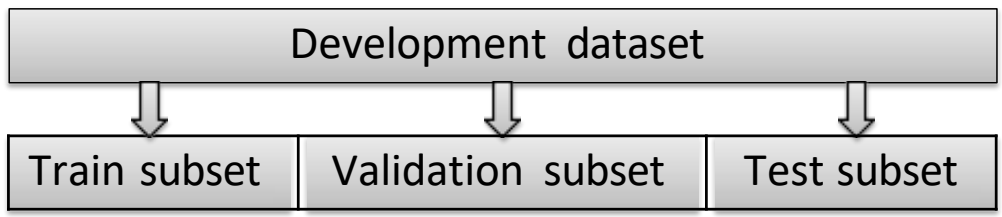
- Use the train subset to train models.
- Use the test subset to evaluate the *final* (ensemble) model.

Unbiased estimation

- Requires:
  - A test subset set aside before starting to experiment.
  - Only a single analysis performed using the test samples.
    - Only a single (ensemble) model.
    - Ideally, also pre-specify what performance metric you would use.
    - It is fine to perform additional analyses but note that these might be biased by the multiple comparisons and learning from previous analyses.
- "if you don't like the results, you have to obtain, and lock away, a completely new test set if you want to go back and find a better hypothesis."
  - *Page 709 in Russell, S. & Norvig, P. Artificial Intelligence: A Modern Approach 3rd edition (Prentice Hall, 2010).*
- Sadly, this is too often ignored.

Validation subset

- A third disjoint subset of the development dataset can be used to evaluate different modelling choices.
  - "Modelling choices": hyperparameters, choices of methods such as type of neural network and optimisation method.

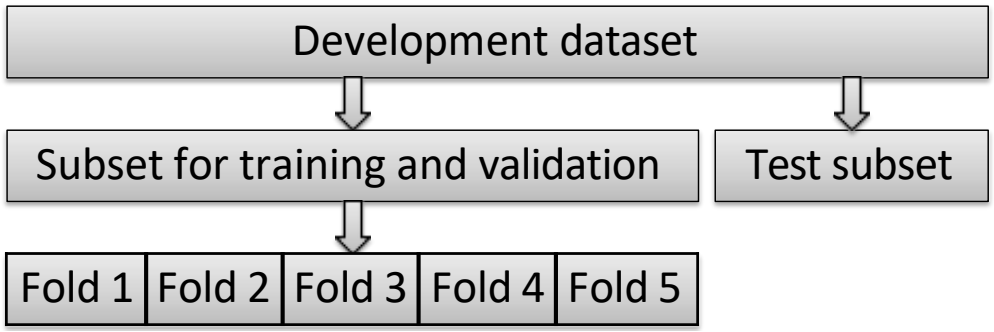


- Nomenclature note: Some use "validation" for testing and "tune subset" as the subset for evaluating modelling choices.

Validation by resampling: k-fold cross validation

- Alternative to fixed validation subset: multiple validation subsets and use average performance to evaluate modelling choices

- k-fold cross-validation:
  1. Randomly partition the non-test subset into k folds of equal size.
  2. Repeat k times:
     Use the k^{th} fold for validation and the other k-1 folds for training.
  3. Use the average of the performance estimates obtained for the validation folds.
     - Could also calculate the standard deviation or other measures of uncertainty.



Validation by resampling: k-fold cross-validation

- Pros:
  - Estimates are less influenced by one particular training.
  - Estimates are less influenced by one particular validation subset.
  - Ends up with k models which can be used in an ensemble model.
- Con:
  - Training time (and validation time) multiplied by k
    - Could be acceptable for reasonably small k, e.g., 5.

Validation by resampling: bootstrapping

- k-fold cross-validation applies a fixed partitioning into folds.
- We can instead repeatedly partition the non-test subset into a train subset and a validation subset. For e.g., using bootstrapping.

- Bootstrapping example: repeat 10 times:
  1. Sample with replacement from the non-test data to obtain a train subset with the same size as the non-test subset.
     - Each train subset will, on average, contain 63.2% of the non-test data but will be the same size as the non-test subset due to duplicates.
  2. Train using the train subset (with duplicates) and use the other non-test data for validation.

Validation by resampling

- Uses the data more effectively than fixed validation subset.
  - Because all non-test data is used for both training and validation.
- Evaluating modelling choices multiple times will make the best performance estimates overly optimistic.
  - Due to multiple comparisons and learning from previous analyses.
  - With a fixed validation subset, the choice that appears to provide best performance might do so only for the particular validation subset.
  - When using a resampling technique, the choice that appears to provide best performance might do so only for the non-test data.
- If data is limited, applying a resampling technique is usually preferable if the added computational expense is acceptable.

Subset sizes

- Common split development data using some percentages, e.g., 70% for training, 15% for validation, and 15% for testing.
- Resampling techniques might allow the use of less validation data.
- Size of the test subset should be considered in relation to the uncertainty of the resulting estimates.

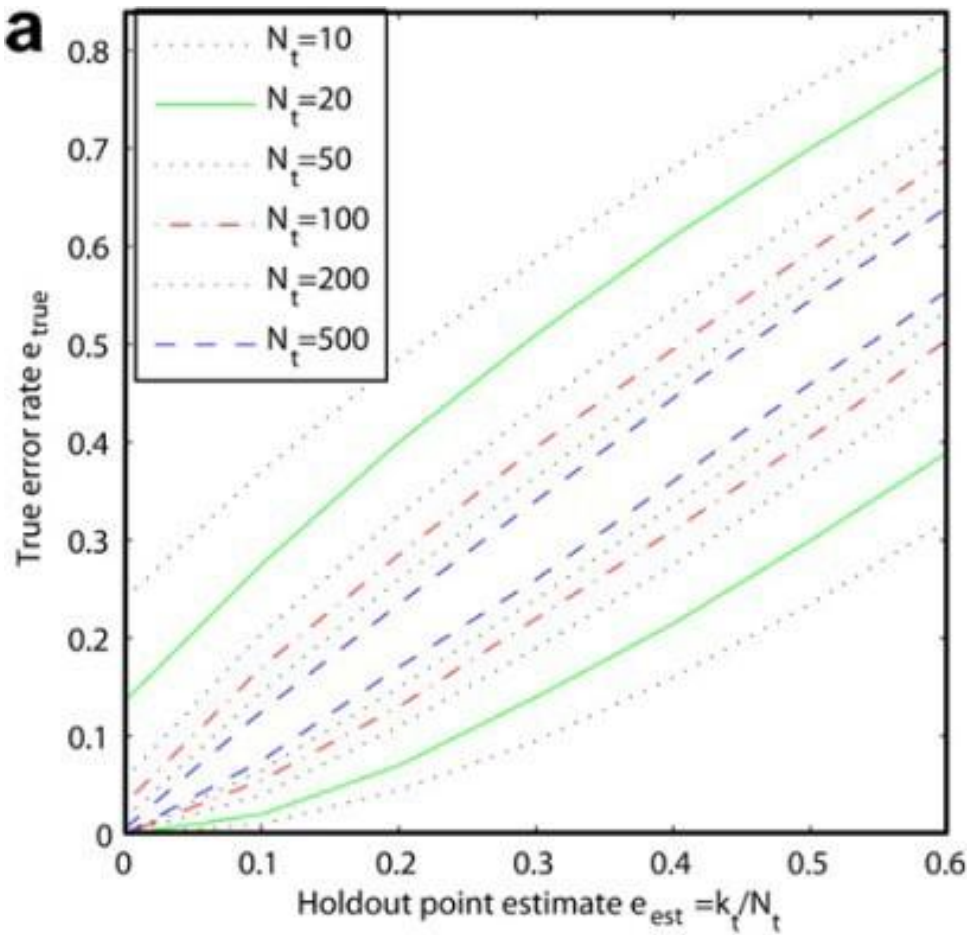


Fig: The lines depict the range of 95% Bayesian confidence intervals. Figure 4a by *Isaksson, A., Wallman, M., Go¨ransson, H. & Gustafsson, M. G. Cross-validation and bootstrapping are unreliable in small sample classification. Pattern Recognit. Lett. 29, 1960–1965 (2008).*

Subset sizes

- Performance ⬌ size of training data
  - Linear on a log-log scale.
  - Performance gained from increasing train subset is initially large and then quickly becomes smaller.

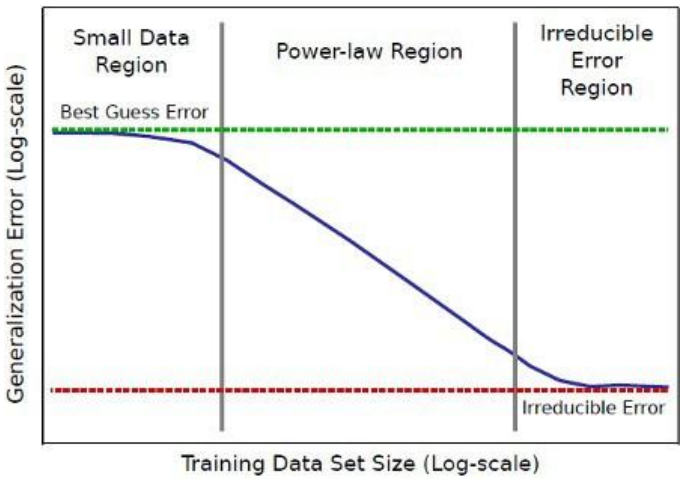
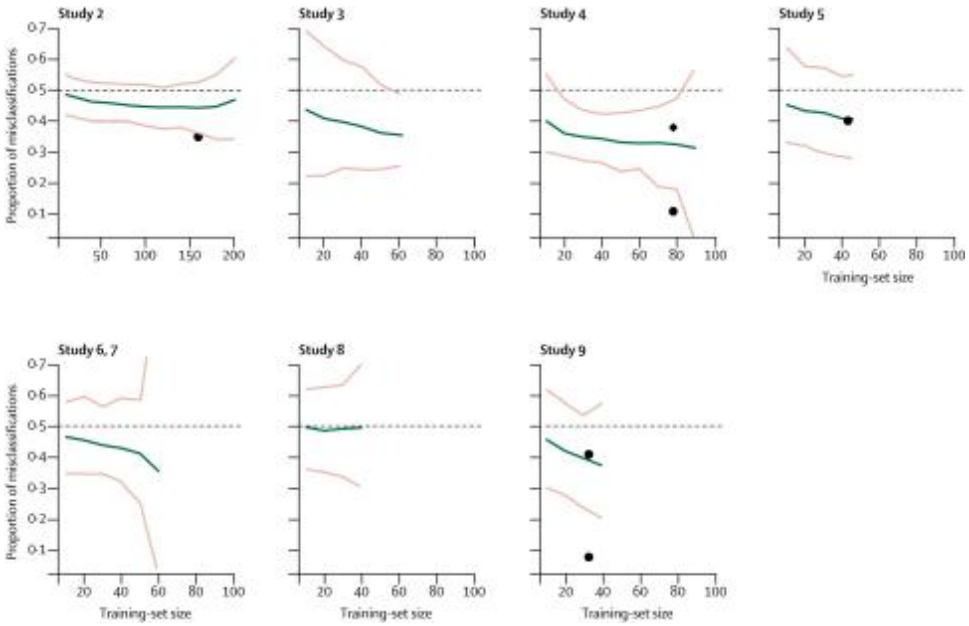


Figure 6: Sketch of power-law learning curves

Hestness, J. et al. Deep learning scaling is predictable, empirically.
https://arxiv.org/pdf/1712.00409.pdf

Validation

- Example from ≈20 years ago, microarray gene-expression profiling
- Michiels et al. used a resampling technique to re-evaluate publicly available data from seven studies that had applied specific validation subsets to evaluate the performance of trained molecular signatures.
  - Studies published in Nature, Nature Medicine, Nature Genetics, Lancet, NEJM, PNAS, and Cancer Cell.
- They found that "Five of the seven studies did not classify patients better than chance."



Proportion of misclassifications in validation sets as a function of corresponding training-set sizes in the seven studies.
Green lines=mean proportion of misclassifications obtained from 500 random training-validation sets as a function of the training-set size. Pale red lines=95% CIs. Dots=misclassification rates in original publications. Iizuka and colleagues published two misclassification rates by two different methods on the same validation set. Diamond=second misclassification rate on a larger independent validation set from the van't Veer study.

Michiels, S., Koscielny, S., & Hill, C. (2005). Prediction of cancer outcome with microarrays: a multiple random validation strategy. The Lancet, 365(9458), 488-492.

Outline

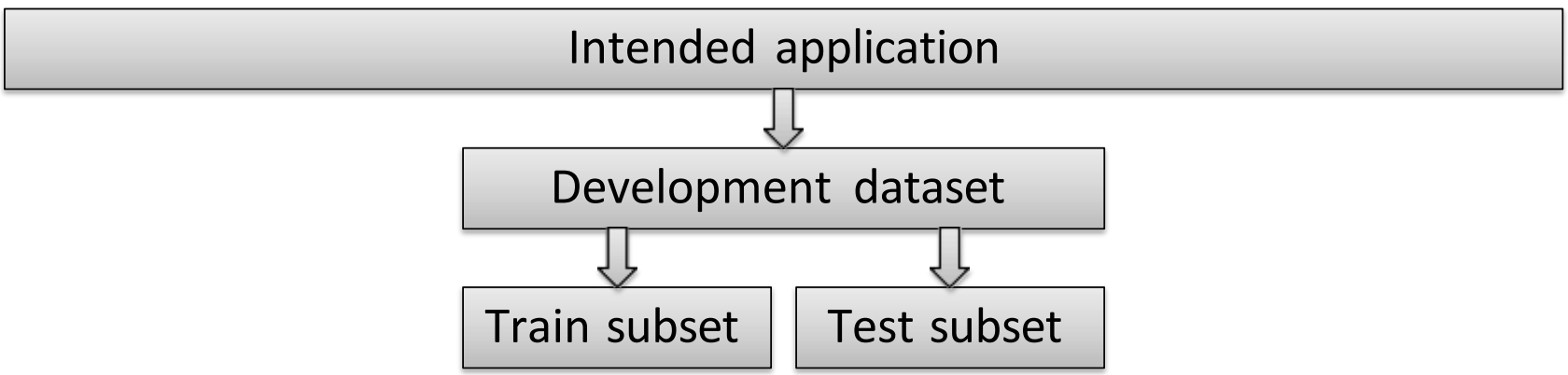
1. Internal testing (recap)
2. External testing
3. Facilitating generalisation
4. Some performance metrics
5. Estimator uncertainty

Representability

- **Recap:** If a random test subset is used for only a single analysis, then the resulting estimate is expected to be unbiased for the development data.
- Does that imply unbiased estimation for the intended application?
- No, not if the development dataset is not representative of the intended application.
- Example: Detecting values of coins
  - Intended application: all coins (with and without scratches).
  - Development dataset: coins without scratches.
  - Consequence: performance on the test subset might be good, but performance in the intended application might be poor.

Representability

- We often assume development data is iid r.v. representative of the intended application.
- Rarely true.



Domain Shift

Training Data



Real world

Domain Shift

Training Data



Real world



Domain Shift

Training Data





Real world



Domain Shift

Training Data



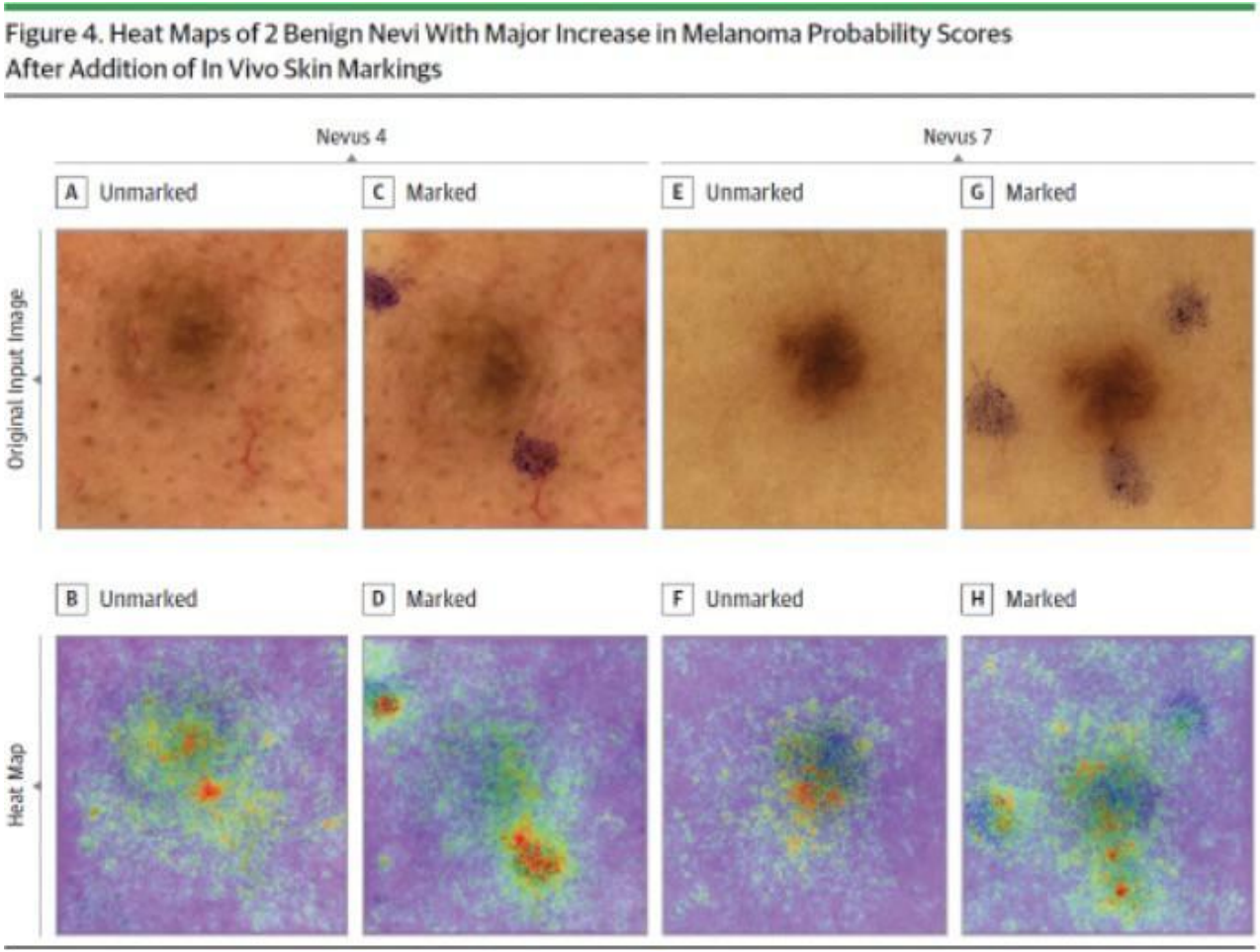
Real world







Representativity



Winkler, J. K. et al. Association between surgical skin markings in dermoscopic images and diagnostic performance of a deep learning convolutional neural network for melanoma recognition. JAMA Dermatol. 155, 1135–1141 (2019).

Representativity

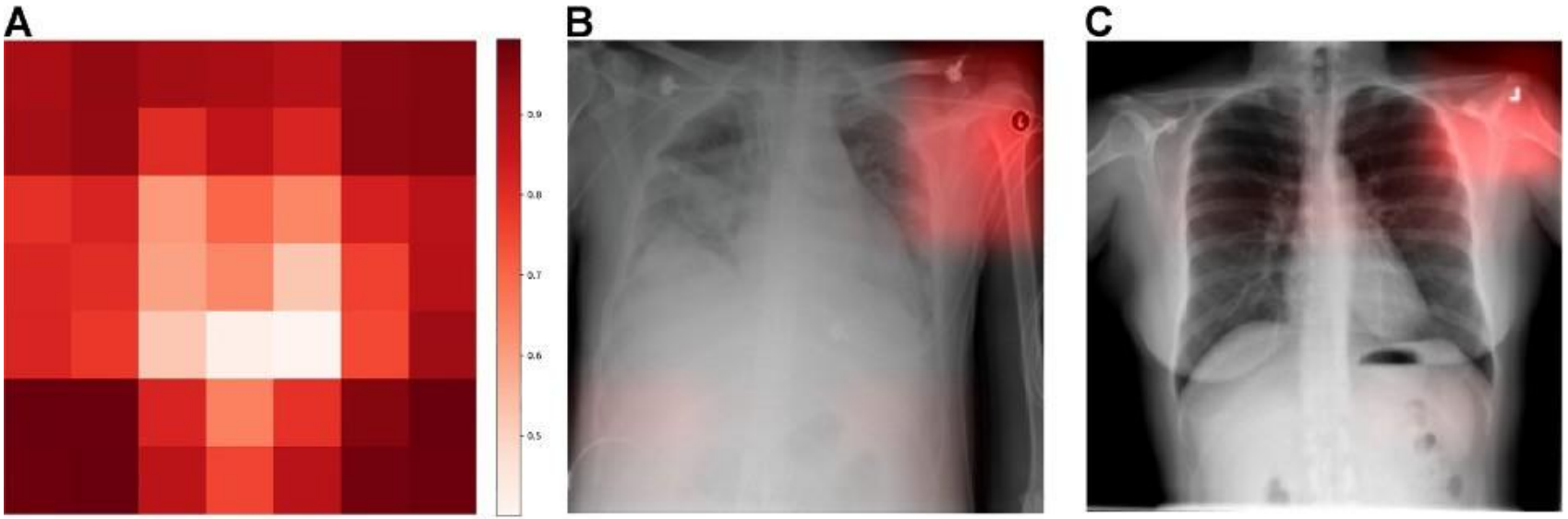


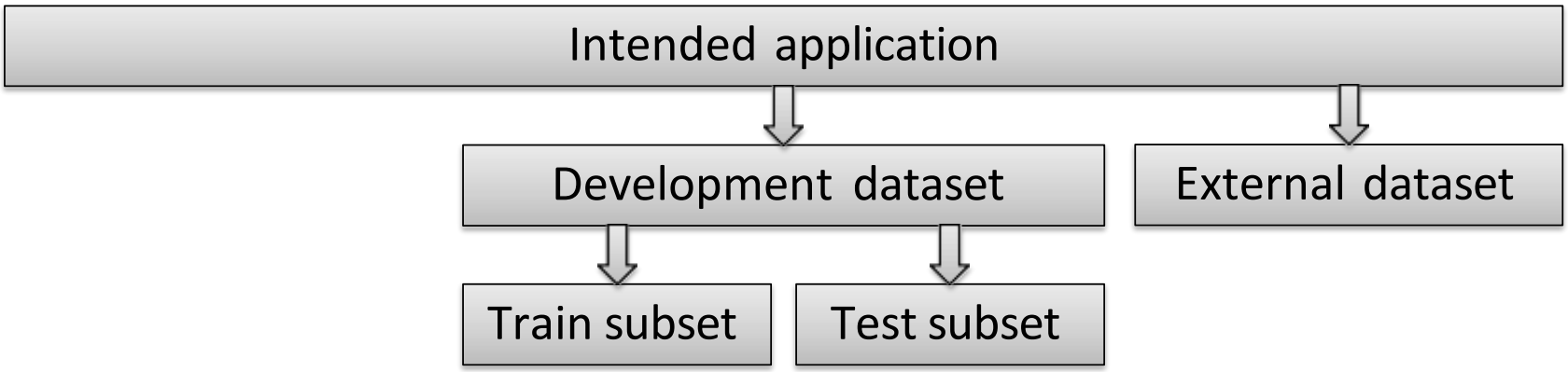
Fig 2. CNN to predict hospital system detects both general and specific image features. (A) We obtained activation heatmaps from our trained model and averaged over a sample of images to reveal which subregions tended to contribute to a hospital system classification decision. Many different subregions strongly predicted the correct hospital system, with especially strong contributions from image corners. (B-C) On individual images, which have been normalized to highlight only the most influential regions and not all those that contributed to a positive classification, we note that the CNN has learned to detect a metal token that radiology technicians place on the patient in the corner of the image field of view at the time they capture the image. When these strong features are correlated with disease prevalence, models can leverage them to indirectly predict disease. CNN, convolutional neural network.

Zech, J. R. et al. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross- sectional study. PLoS Med. 15, e1002683 (2018).

Representability

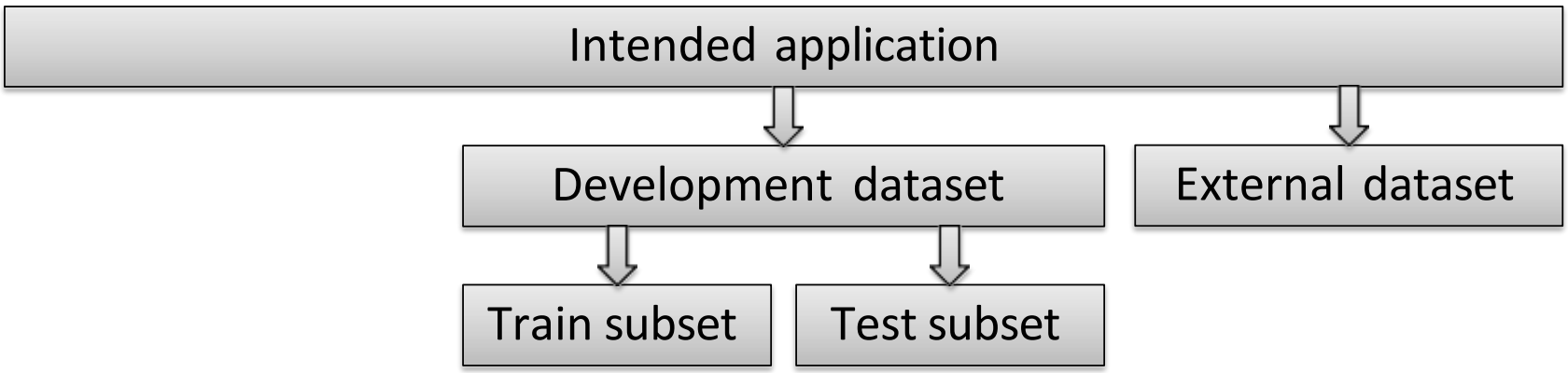
Data external to the development dataset are needed to obtain **more realistic performance estimates**

- External: e.g. geographical locations, temporally different (e.g. from a later time period)
- Note: External datasets are a luxury that is too seldom available.



Representability

- External dataset usually does not fully represent the intended application.
- Good performance on an external dataset:
  - Indicates good generalisation to one of the multiple intended settings.
    - Assuming the external dataset is representative of an intended setting.
  - Suggests that the model would also generalise to other settings



Use of external datasets

- The external dataset should be used only once to evaluate the final model.

Use of external datasets

- External datasets are becoming more common in applied deep learning studies.
- Example from cancer diagnostics:

Statistics for studies with ≥20 citations per year or published in journals with impact factor ≥10.

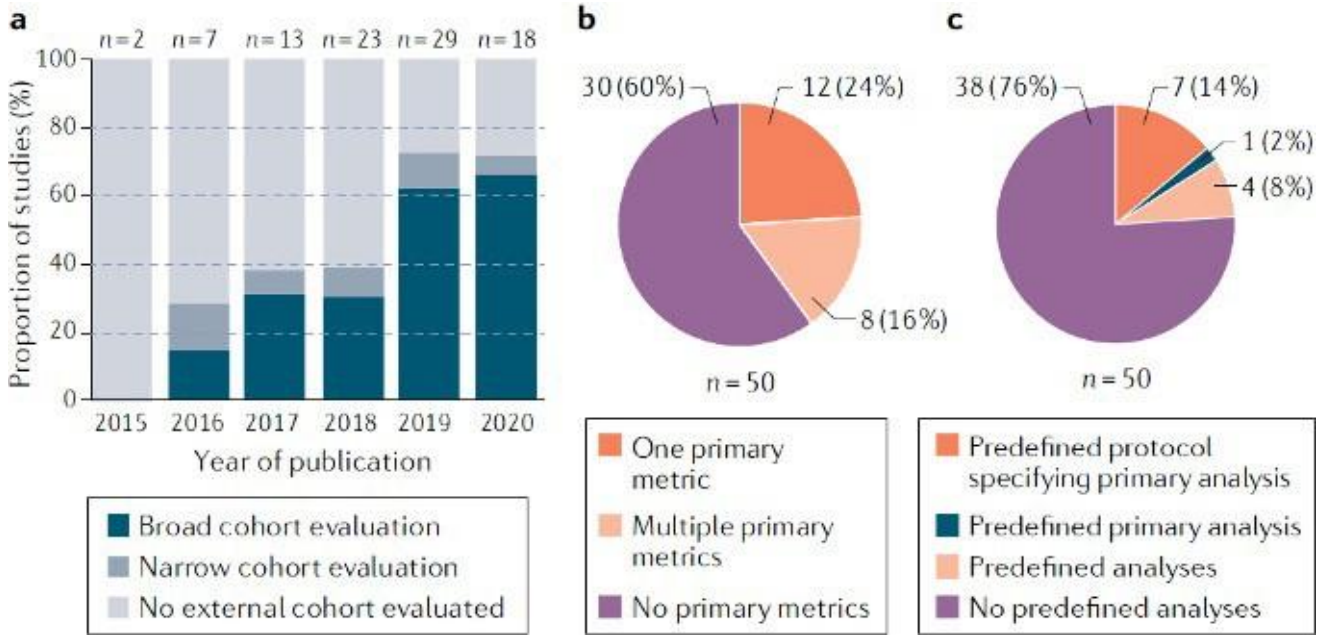


Fig 1 in Kleppe, A. et al. Designing deep learning studies in cancer diagnostics. Nat. Rev. Cancer 21, 199–211 (2021)

- Pre-specified analyses of the external test is rarely reported.
  - Except in clinical trial studies, where pre-specification is usually required and publicised in advance of patient recruitment.

Outline

1. Internal testing (recap)
2. External testing
3. Facilitating generalisation
4. Some performance metrics
5. Estimator uncertainty

Facilitating generalisation

Control the network's capacity.

- Reduce # trainable parameters
- Dropout
- Weight decay

Facilitating generalisation

- Control the network's capacity.
- Facilitate learning.
  - Residual connections.
  - Batch normalisation.
  - Transfer learning.
  - Learning rate schedule.
  - Optimisation method.

Facilitating generalisation

- Control the network's capacity.
- Facilitate learning.
- Image-specific normalisation.
  - Aims at making different images more similar.
- Data augmentation.
  - E.g., randomly distort brightness, colour, contrast, rotation, sharpness, or size of each input image during training.
  - See, e.g., Shorten, C. & Khoshgoftaar, T. M. A survey on image data augmentation for deep learning. J. Big Data 6, 60 (2019).
- More diverse training data.

Facilitating generalisation

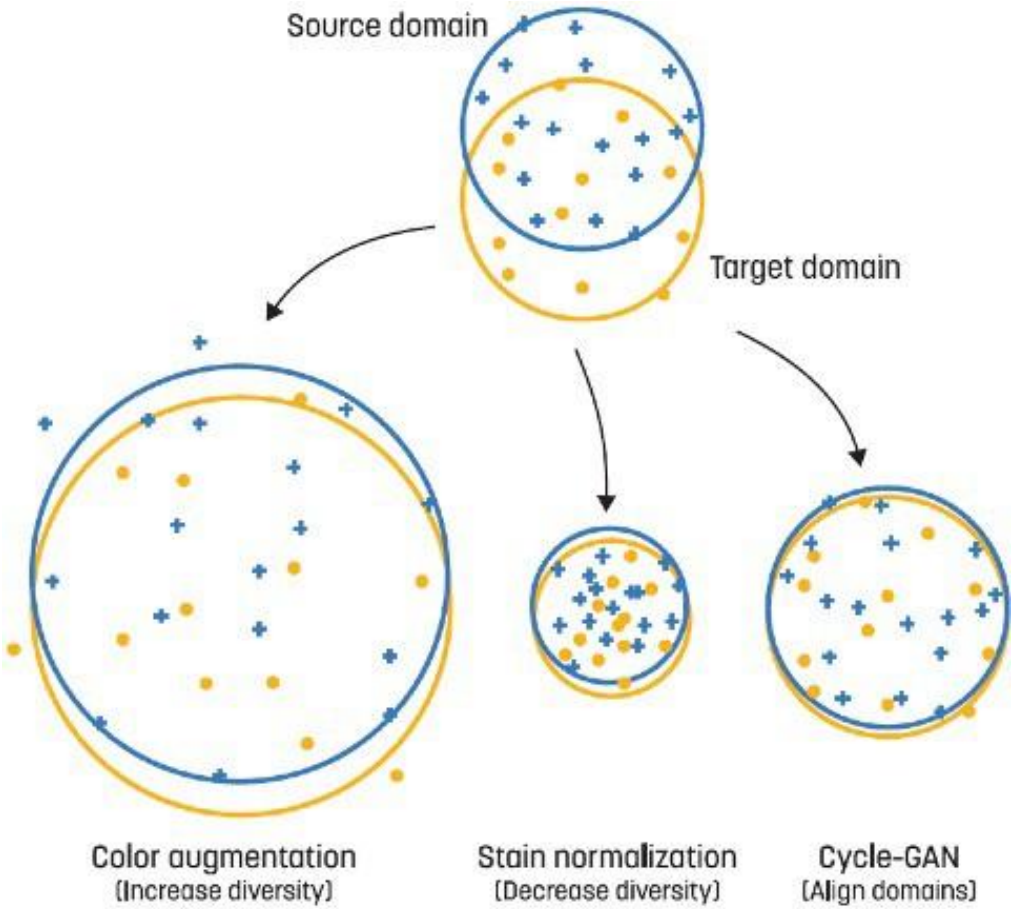


Fig. 6. Data preparation aim at increasing the intersection between domains. Different data transformation techniques do this with different strategies.

Stacke, K., Eilertsen, G., Unger, J. & Lundstr¨om, C. Measuring domain shift for deep learning in histopathology. IEEE J. Biomed. Health Inform. 25, 325–336 (2021).

Normalisation

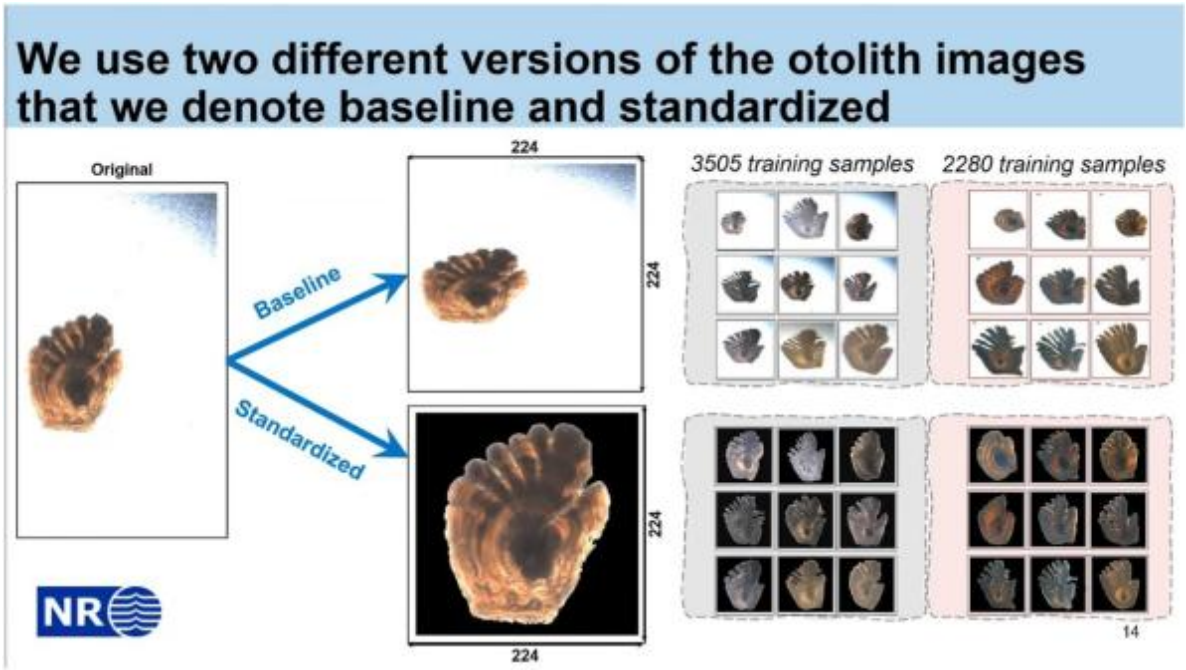
- Dataset-based input normalisation can facilitate faster training convergence, possibly leading to a better model.
  - All images are transformed using the same function and parameters.
- Image-specific normalisation can make images more similar.

  Can apply parameters extracted from that particular image.

  Simple example: tf.image.per image standardization
  - Scale by the image mean and std
  - Sounds very similar to dataset-based input normalisation, but note that the mean and standard deviation are computed for the particular image and not across all training images.

Normalisation

Another example of image-specific normalisation is cropping and centre alignment of the region of interest.



153 test samples

Target domain	Lower Bound	CoGAN (Average over 5 runs)	Upper Bound
Baseline	5.0319	2.5332 (74.5%)	1.6764
Standardized	5.8589	2.0889 (93.6%)	1.8311

RMSE (year)

Ordon˜ez, A. Deep Domain Adaptation Applied to Fish Age Prediction. Visual Intelligence Seminar (2021).
https://www.youtube.com/watch?v=5NwGGsHyqbA

Augmentation: diverse training data

- The natural variability in the training data matters.
- Artificial and natural data variation complement each other.

Outline

1. Internal testing (recap)
2. External testing
3. Facilitating generalisation
4. Some performance metrics
5. Estimator uncertainty

Performance metrics: which to choose?

- Often chosen by convention
- A model for practical usage should primarily be evaluated using a performance metric that **reflects the intended application**
  - If the end user of your model will simply use the classifications, then the performance metric should also only use the classifications (not the continuous prediction scores).
    - Note that metrics calculated from classifications are often more conservative than metrics calculated from prediction scores.
  - If this is not done, then it can be difficult to interpret how well the model will perform in the intended application, both in itself and relative to other models.
- Some ignore relevant aspects and should be complemented by other statistics.

Confusion matrix

- Each element specifies the number of samples with a particular class c_A predicted to be a particular class c_P

		Predicted class			
		c_1	c_2	…	c_k
Actual class	c_1	$n_{1,1}$	$n_{1,2}$	…	$n_{1,k}$
	c_2	$n_{2,1}$	$n_{2,2}$	…	$n_{2,k}$
	…	…	…	…	…
	c_k	$n_{k,1}$	$n_{k,2}$	…	$n_{k,k}$

c_i specifies a class and $n_{a,b}$ specifies a count

- Diagonal elements specify correctly classified samples.

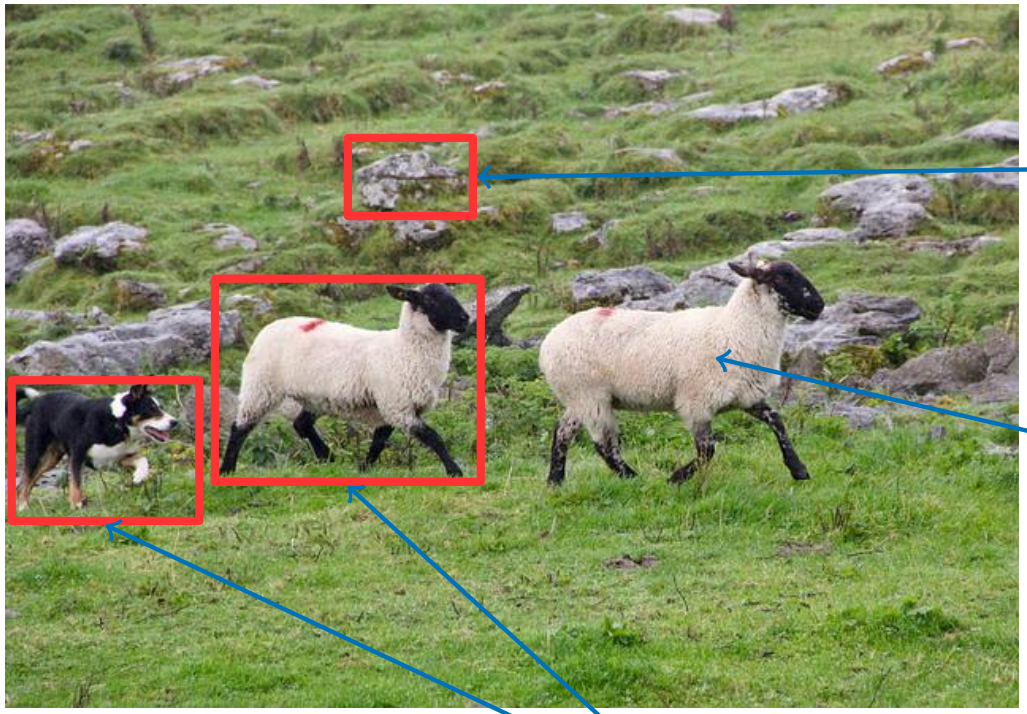
Confusion matrix

In the special case of two classes, the confusion matrix contains only four values, often named as:

		Predicted class	
		Predicted positive	**Predicted negative**
Actual class	**(Actual) positive**	True positive (TP)	False negative (FN)
	(Actual) negative	False positive (FP)	True negative (TN)

Useful in object detection

Assume a model is used to detect objects of a single class “animal”



Nomenclature

- FP = a Type I error
- FN = a Type II error

Accuracy

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

- Top-1 accuracy: how often is the model's highest prediction correct?
- Top-5 accuracy: how often is one of the model's top 5 highest predictions correct?

- When is accuracy misleading?
  - Imbalanced datasets
  - Doesn't distinguish between confident and uncertain predictions
  - Doesn't consider similar classes for fine-grained classification

Precision and Recall

Can be deduced from the confusion matrix for two classes.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Precision = Accuracy of positive predictions

$$\text{Recall} = \frac{TP}{TP + FN}$$

Recall = Percentage of positive elements found

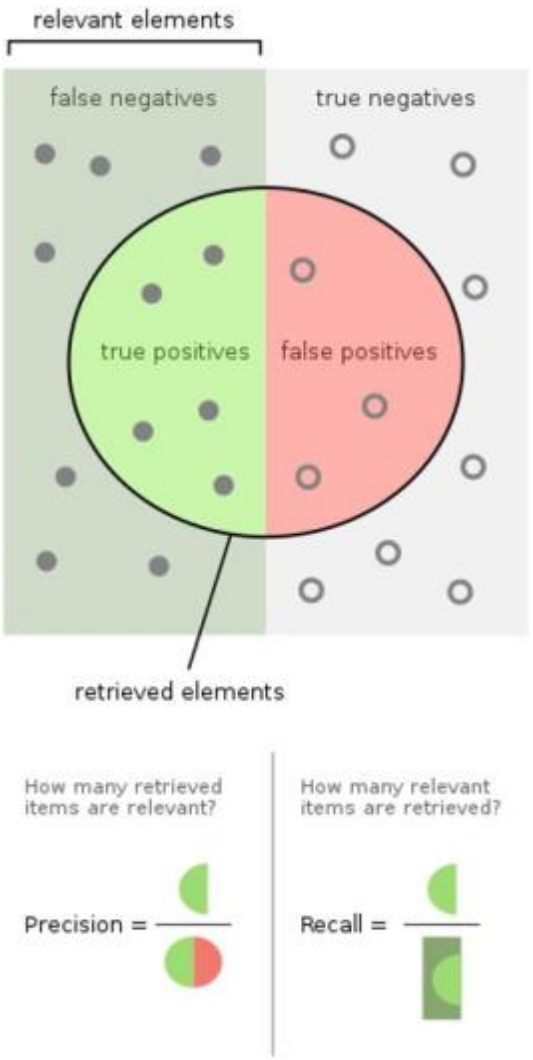
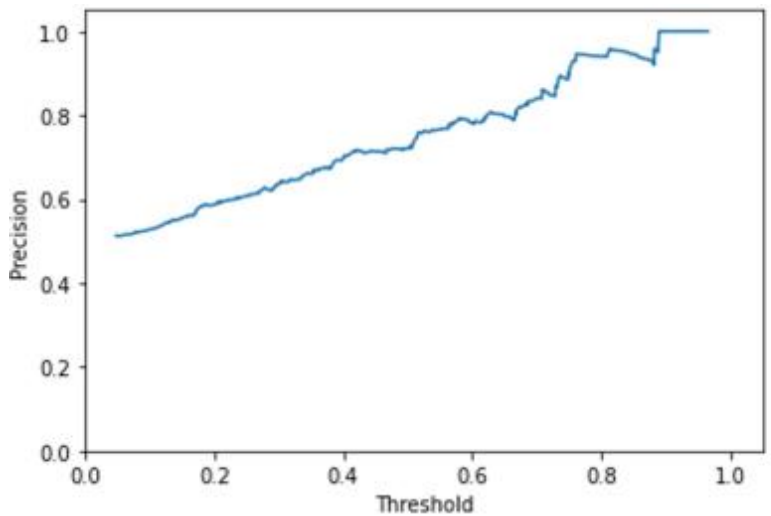
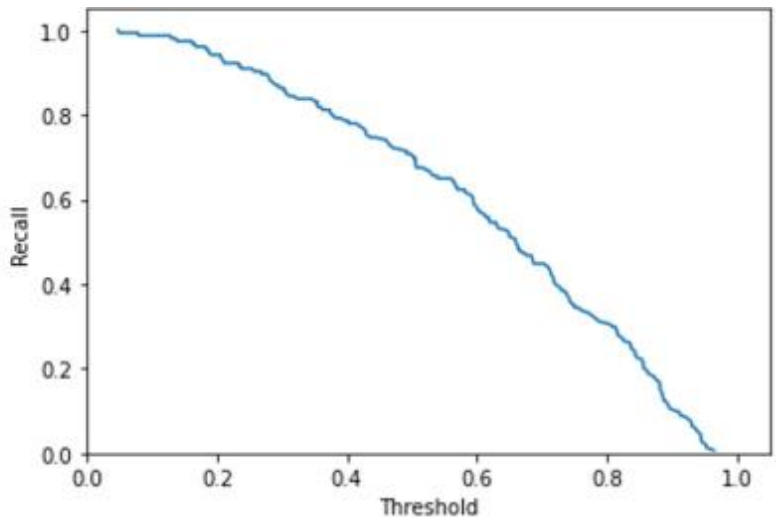


Image source: By Walber - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=36926283

Precision and recall

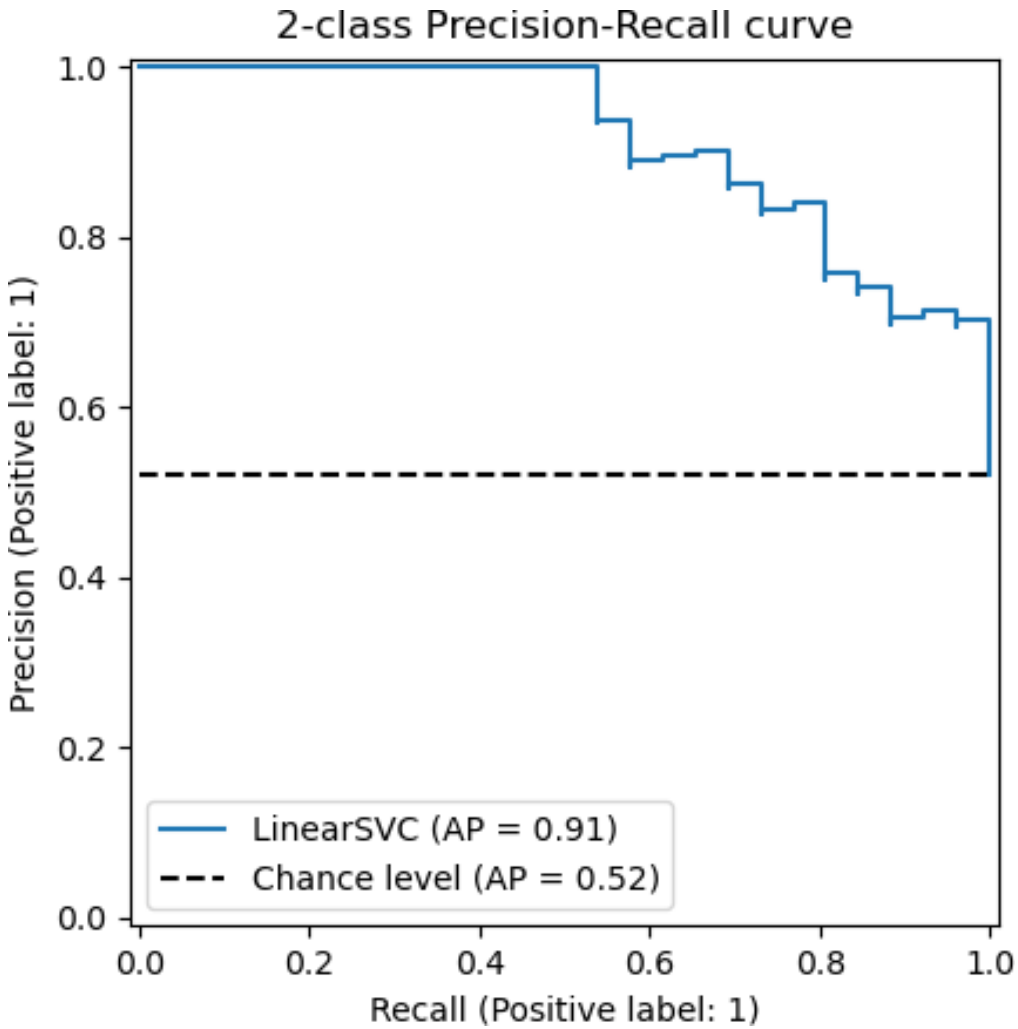
For a trained binary classifier:
As you increase the threshold for classifying the input as positive, precision will typically go up, and recall will typically go down.





Precision-recall curve

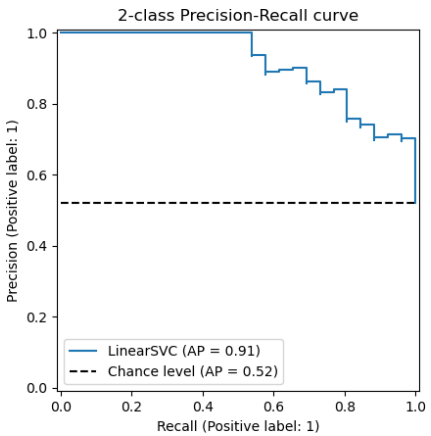
A curve depicting the pairs of precision and recall for different thresholds of the prediction score of the positive class.



https://scikit-learn.org/stable/_images/sphx_glr_plot_precision_recall_001.png

Precision-recall curve

1. Compute positive prediction scores for each sample in the evaluation set.
2. Start with a threshold larger than all scores.
   - Gives no positive predictions, so recall is 0 and precision is undefined (the expression is 0/0, can define it as 1).
3. Repeat until the threshold is below all scores:
   1. Lower the threshold to include one more of the observed scores.
      - Note that multiple samples may have this score.
   2. Calculate the recall and precision when using the new threshold.
   3. Draw a curve between the new and the previous (recall, precision) pair.
      - There are multiple choices for how to do this.
      - In scikit-learn above 0.19: Draw a horizontal line between the new and previous recall at the new precision and then a vertical line between the new and previous precision at the previous recall.



https://scikit-learn.org/stable/_images/sphx_glr_plot_precision_recall_001.png

Precision-recall curve

Score	GT
0.1	0
0.2	1
0.3	0
0.6	1
0.9	1

	Precision	Recall
0	3/5=0.6	3/3=1
0.2	¾=0.75	3/3=1
0.3	2/3=0.67	2/3=0.67
0.6	2/2=1	2/3=0.67
0.9	1/1=1	1/3=0.33
1	0/0==1	0/3=0

Average precision (AP)

- The area under the precision-recall curve is called the average precision (for the positive class)
- Measures how well the prediction scores match the class labels
  - Higher is better; the AP of a perfect classifier is 1
  - AP for a random classifier is equal to the proportion of actual positives in the evaluation set
- Some compute the area under an interpolated version of the precision-recall curve, but then depends on how a curve is drawn between adjacent (recall, precision) pairs

Average precision (AP) (not relevant for the exam)

An alternative approach to computing AP:

1. Compute positive prediction scores for each sample in the evaluation set.
2. Sort samples by their scores in descending order (i.e., sample with the highest score first).
3. Secondary sort on ascending label, implying that if multiple samples have the same score, then actual negative samples should come before actual positive samples with the same score.
4. Calculate:

$$AP = \frac{1}{R} \sum_{k=1}^{N} y_k P@k$$

- R is the number of actual positives in the evaluation set:

$$R = \sum_{k=1}^{N} y_k = TP + FN$$

- N is the number of samples in the evaluation set.
- y_k is 1 if sample k (i.e., sample with index k after sorting) is an actual positive; otherwise, 0.
- $P@k$ is precision at k, i.e., the precision for the first k sample after sorting:

$$P@k = \frac{1}{k} \sum_{i=1}^{k} y_i$$

Mean average precision (mAP)

- If there are more than two classes, then the average precision can be calculated per class.
- “Positive” is the relevant class and “Negative” is all other classes.
- Perform the calculation for all classes.
- The arithmetic mean of the average precisions is the mAP:

$$mAP = \frac{1}{k}\sum_{i=1}^{k} AP_i$$

where k is the number of classes

- Arithmetic means of other sets of average precisions are also referred to as mAP.

F1 score

- For a fixed threshold, compute precision and recall, can be combined to an F-score
- $F_1 = \frac{2}{\text{Recall}^{-1}+Precision^{-1}} = 2\frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2\,\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}$
- $Precision = \frac{TP}{TP+FP}$. $\text{Recall} = \frac{TP}{TP+FN}$
- The harmonic mean of Precision and Recall: symmetrically represents both
- F_β : values one metric more than the other.
    - Attach β times as much importance to recall as precision

Sensitivity and specificity

Can be deduced from the confusion matrix for two classes.

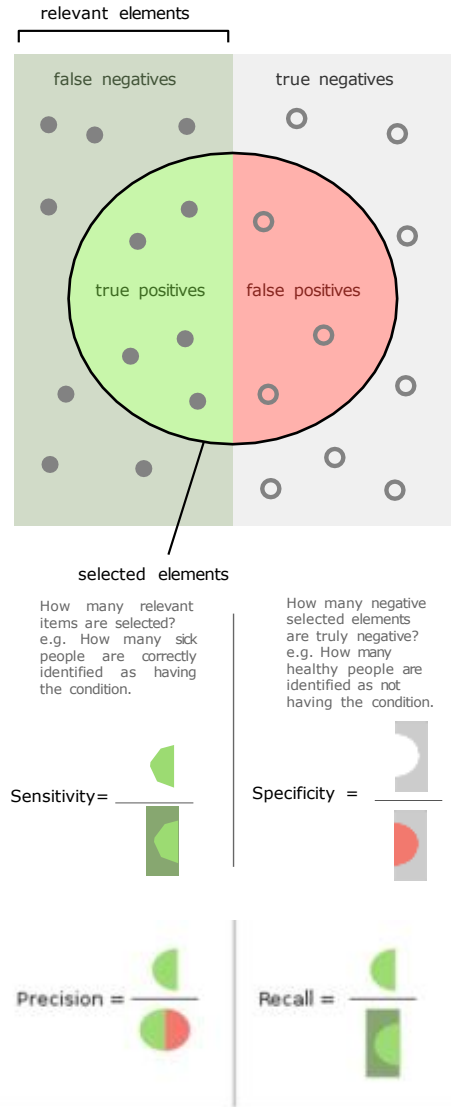
$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

- Proportion of actual positives that are positive predictions.
- Same as recall (a.k.a. true positive rate).

$$\text{Specificity} = \frac{TN}{TN + FP}$$

- Proportion of actual negatives that are negative predictions.
- A.k.a. true negative rate.
- False positive rate = 1 - Specificity

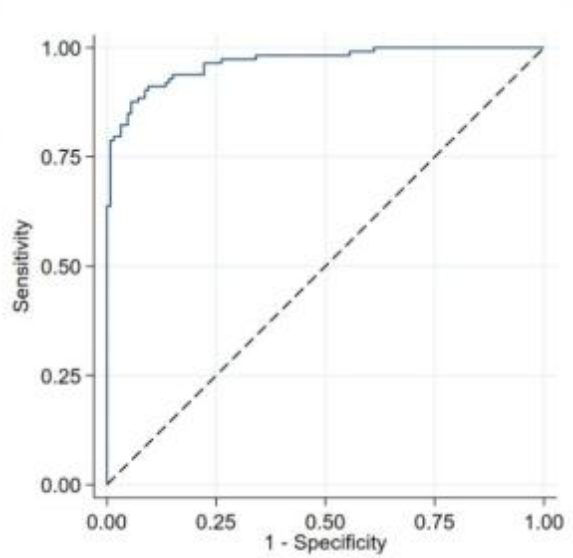
Source:https://en.wikipedia.org/wiki/File:Sensitivity_and_specificity_1.01.svg



Receiver Operator Characteristic (ROC) curves

ROC curves can be created similarly as precision-recall curves, except that:

- Using each threshold, the sensitivity and specificity are calculated and plotted instead of the recall and precision.
- Linear interpolation is used between the new and the previous (1-Specificity, Sensitivity) pair.
- ROC curves start at (0,0) and end at (1,1).



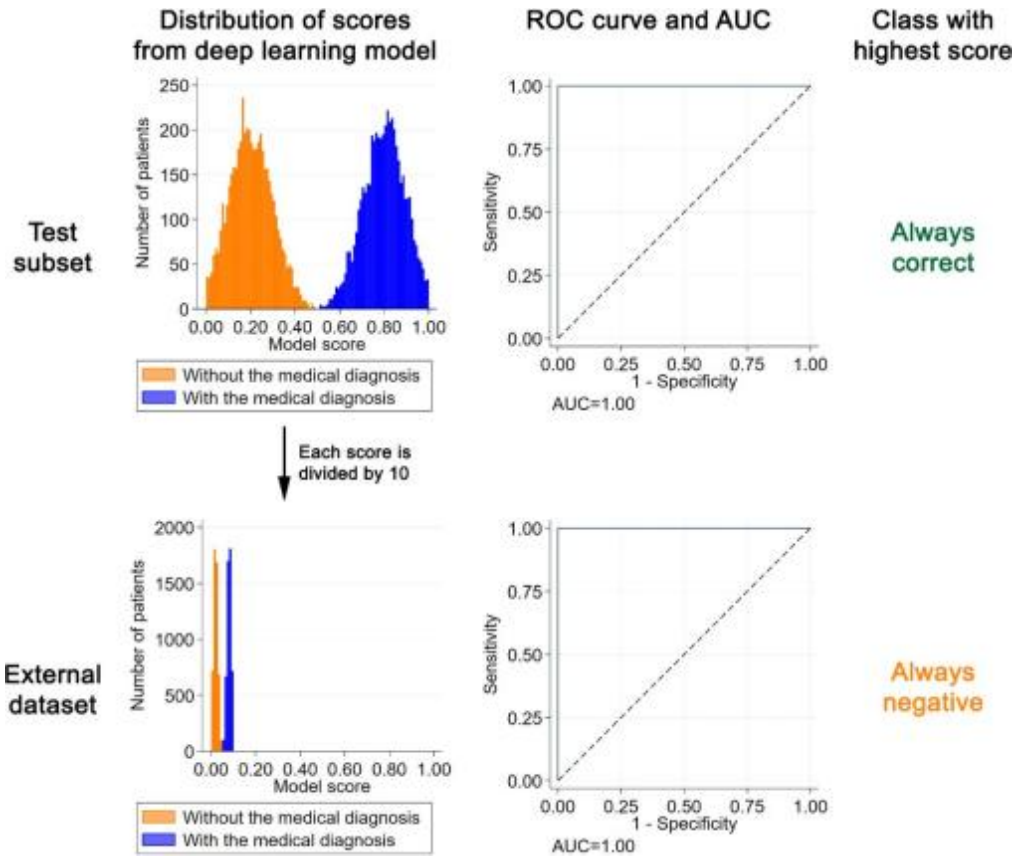
Area under the curve (AUC)

- AUC and AUROC is the area under the ROC curve.
- Create all possible pairs with one element chosen from the positive class and one from the negative class. It can be shown that AUROC is the **proportion of pairs** where the prediction score of the positive class > prediction score of the negative class.
  - If the scores are identical, then 0.5 is added (instead of 1 or 0) to the calculated proportion's nominator.
- Thus, the AUROC can be viewed as the probability that the predicted score for the positive class is larger for an actual positive sample than for an actual negative sample.

AP and AUROC

- Summarise the discriminatory value of the prediction scores.
- Only use the ranking of the prediction scores, not their absolute values.
- Are thus invariant to any strictly increasing transform.
  - That is, the metric value of the transformed prediction scores will be the same as the metric value of the original prediction scores.
- A concern when combined with external testing.
  - A distribution shift between the development dataset and the external dataset can result in changes in the score distribution that resemble those caused by some strictly increasing transform.
- AP and AUROC may thus **hide a type of model bias that is not uncommon**.

AP and AUROC



What if the scores of a second external dataset were transformed by $s^{1/10}$ compared to the test subset?

- AUC will still be 1.00, but the class with the highest score will nearly always be positive (specifically if $s > 0.5^{10} \approx 0.00098$).

AP and AUROC

- To overcome the shortcomings of AP and AUROC, one could use any other metric (like balanced accuracy) along with AP/AUROC.
- The other metric could reveal any biases and shortcomings that AP or AUROC hide.

Balanced Accuracy

		Predicted class: Predicted positive	Predicted class: Predicted negative
Actual class	**(Actual) positive**	True positive (TP)	False negative (FN)
	(Actual) negative	False positive (FP)	True negative (TN)

- Accuracy = $\frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}}$
- Recall / Sensitivity / True Positive Rate (TPR) = $\frac{\mathrm{TP}}{\mathrm{TP+FN}}$
- Specificity / True Negative Rate (TNR) = $\frac{\mathrm{TN}}{\mathrm{TN+FP}}$
- Balanced Accuracy = $\frac{\mathrm{TPR+TNR}}{2}$
- Balanced accuracy is useful when classes are imbalanced
- When can it be misleading?

Balanced Accuracy

Actual class \ Predicted class	c_1	c_2	…	c_k
c_1	$n_{1,1}$	$n_{1,2}$	…	$n_{1,k}$
c_2	$n_{2,1}$	$n_{2,2}$	…	$n_{2,k}$
…	…	…	…	…
c_k	$n_{k,1}$	$n_{k,2}$	…	$n_{k,k}$

c_i specifies a class and $n_{a,b}$ specifies a count

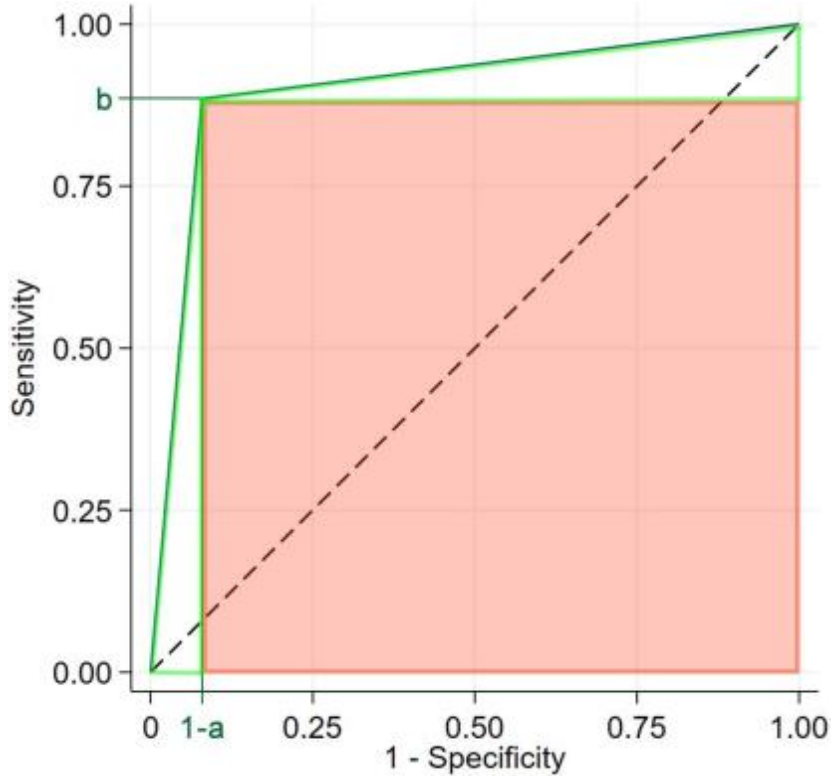
- Accuracy = $\frac{n_{11}+n_{12}+\ldots+n_{kk}}{(n_{11}+n_{12}+\cdots+n_{1k})+(n_{21}+n_{22}+\ldots+n_{2k})+\ldots+(n_{k1}+n_{k2}+\ldots+n_{kk})}$
- Recall / Sensitivity / of c_1 = S_{c1} = $\frac{n_{11}}{n_{11}+n_{12}+\ldots+n_{1k}}$
- Similarly, calculate the sensitivity of all the classes
- Balanced Accuracy = $\frac{S_{c1}+S_{c2}+\ldots+S_{ck}}{k}$

Performance metrics

- For many classification tasks, the ultimate goal is to simply predict the correct class.
- If so, a classification metric such as balanced accuracy may reflect the intended application.
- Any metric calculated using classifications rather than prediction scores will give a lower performance estimate in the case of the aforementioned type of model bias.
  - Assuming the way that classifications are obtained from prediction scores is not influenced by the external dataset.
    - This should, therefore, be pre-specified, e.g., as “select the class with largest prediction score” or “apply these thresholds” (which might be thresholds identified as suitable using the development dataset).

Performance metric (not relevant for the exam)

- Also, the AUROC calculated from classifications instead of prediction scores is a reasonable metric.
- It is actually equivalent to the balanced accuracy in the case of two classes:
  - Let *b* be the sensitivity and *a* the specificity of a dichotomous classifier (see figure above).
  - The AUROC is the sum of the area of one rectangle and two triangles.



AUROC is the area of the red rectangle + the areas of the two green triangles

$$AUC = (1-(1-a))b + \frac{(1-a)b}{2} + \frac{(1-(1-a))(1-b)}{2}$$

$$= ab + \frac{b-ab}{2} + \frac{a-ab}{2} = \frac{a+b}{2}$$

Outline

1. Internal testing (recap)
2. External testing
3. Facilitating generalisation
4. Some performance metrics
5. Estimator uncertainty

Uncertainty of performance estimate

- A performance estimate using a finite set of data is always associated with uncertainty
  - Technically, the estimator (i.e., the rule for calculating the estimate) has a distribution around the true performance.
- This uncertainty should be quantified and reported with the estimate of the performance metric.
- Common to estimate a 95% confidence interval.
  - A procedure for creating 95% confidence intervals for a parameter (e.g., a performance metric) should be such that, on average, 95% of created confidence intervals contain the true value of the parameter.
  - Often misinterpreted as “95% probability of containing the true value”. But given a concrete confidence interval, the interval either contains the true value or not (it is no longer a matter of probability).

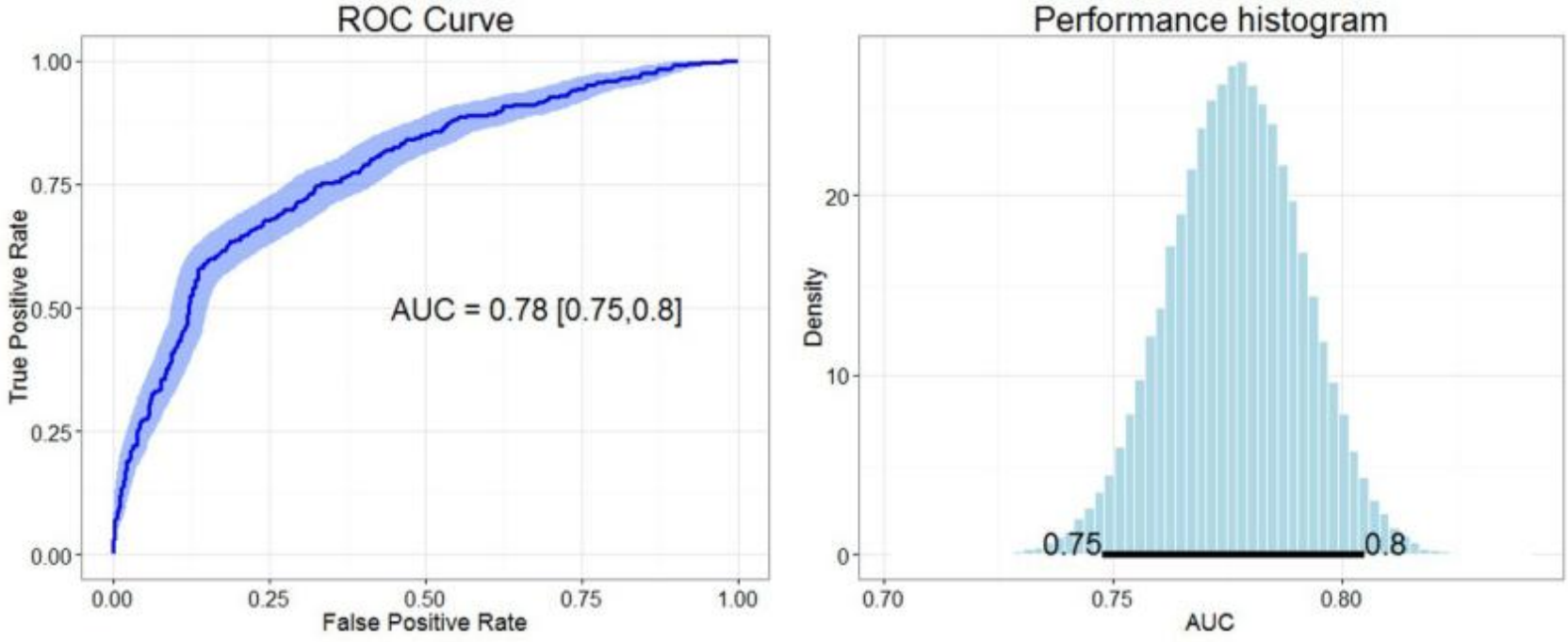
Uncertainty of performance estimate

- For many performance metrics, there exist specific procedures for estimating confidence intervals.
  - E.g. for sensitivities, one of the formulas for binomial proportion confidence interval could be applied, e.g. Clopper-Pearson interval or normal approximation interval.
- Bootstrap confidence intervals are common and can be calculated for (almost?) all metrics.
  - Although bootstrap confidence intervals might be a bit narrow because of the similarity between the sampled sets.
    - *Isaksson, A., Wallman, M., G¨oransson, H. & Gustafsson, M. G. Cross-validation and bootstrapping are unreliable in small sample classification. Pattern Recognit. Lett. 29, 1960–1965 (2008).*

Bootstrap confidence interval

Create a distribution of the parameter by repeating, e.g., 10000 times:

1. Sample with replacement from the original test set to obtain a resampled test set with the same size as the original test set.
2. Compute the parameter of interest using the resampled test set.



https://i2.wp.com/www.epeter-stats.de/wp-content/uploads/2015/05/example-1.png

Uncertainty of performance estimate

If the model performs over-optimistically on the test set, then the confidence interval will inherit this over-optimism.

- This may also happen for external datasets that are used for model selection through testing multiple options and selecting the best one.

Key learning goals

- Design projects in a way that makes the performance estimation as reliable as is practically feasible
- Think critically about the reliability of evaluations of a trained neural network
- Be able to suggest approaches for facilitating generalisation, in particular, generalisation to external data
- Understand AP and AUROC and the limitations of performance metrics that use only the ranking of prediction scores
- Interpret a performance estimate as a value with associated uncertainty and be able to estimate this uncertainty