Evaluation of AI models

Krzysztof J. Geras

Opening notes

- The only purpose of this talk is that you learn stuff.
- If you have questions, ask. It’s more interesting for everyone that way, including me.
- We can go through the slides or we can stop and focus on what you find most interesting.
- It’s better to develop a understanding of fewer things than to have a shallow understanding of many things.
- My goal is to give an idea for what is possible and enable you to self-study effectively.
- We will focus on classification.
- This is probably the most important topic at this summer school for you.

There is a lot of bad science :(

Machine Learning in one slide

(training)
data → ? → predictions
(on the test set)

learning
and hyperparameter selection (validation set)

- Supervised learning
- Unsupervised learning
- Reinforcement learning
- …

Classification

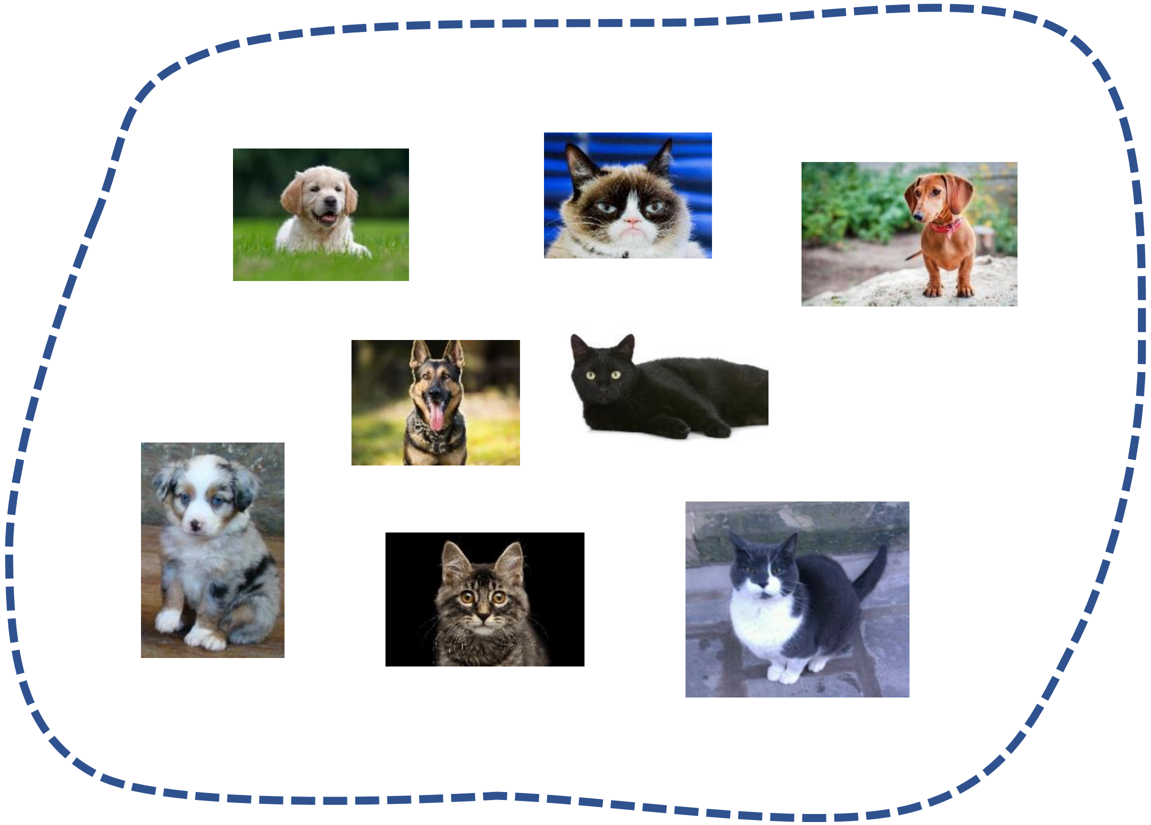
$$\boxed{f} : \boxed{X} \to \boxed{\{0, 1, \ldots, k\}}$$

a function

an arbitrary set

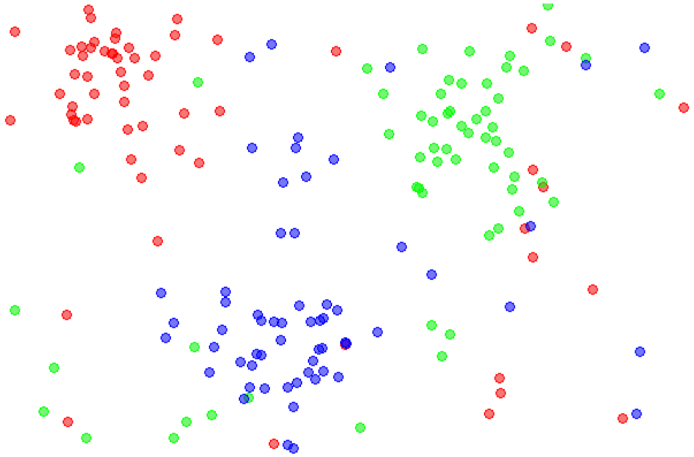
a discrete set

Classification

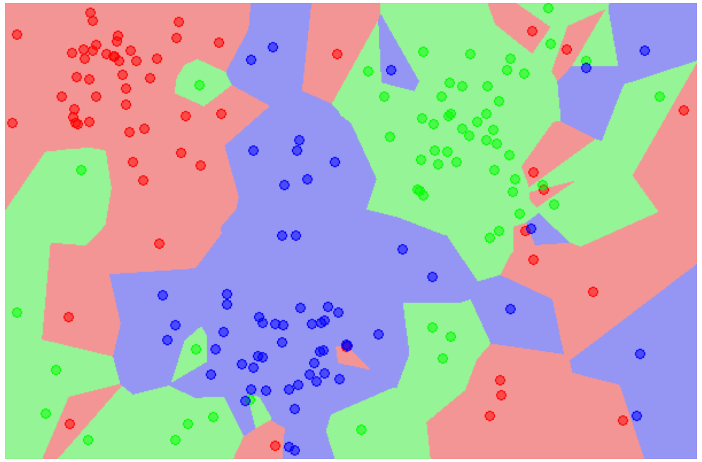


→ {dog, cat}

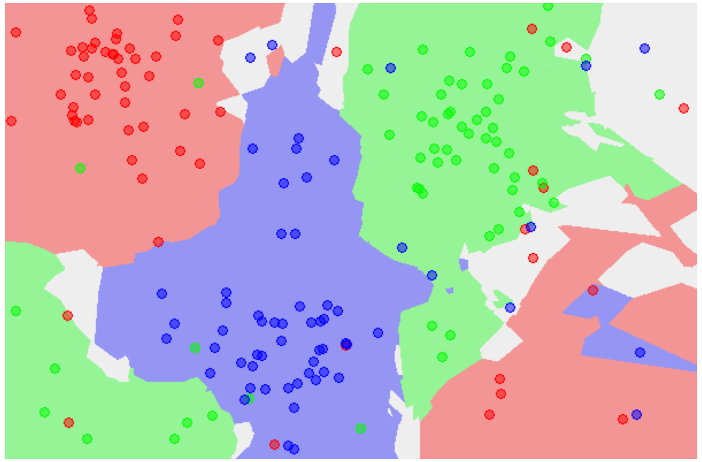
k-nearest neighbours



data



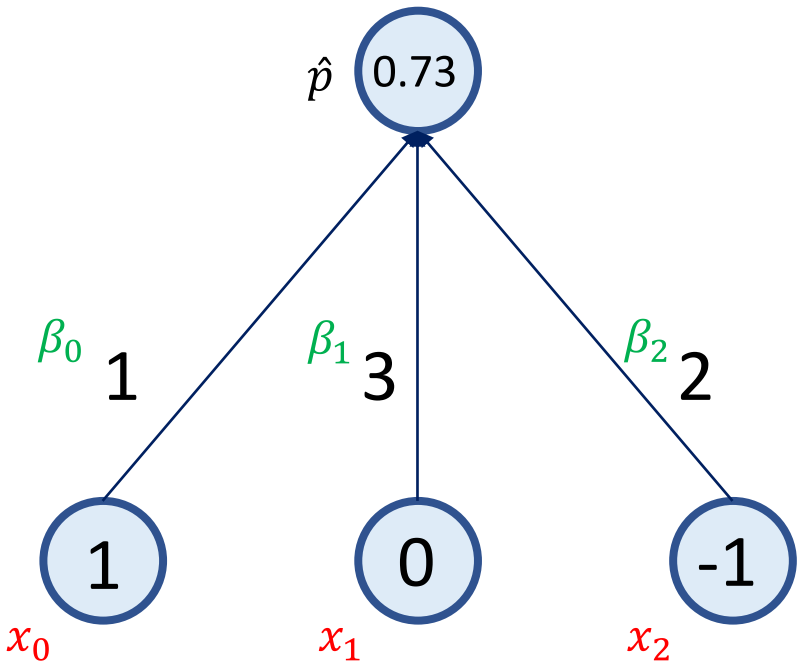
1-nn



5-nn

No parameters! Hyperparameters!

Logistic regression



$$\log \frac{\hat{p}}{1-\hat{p}} = \beta_0\, x_0 + \beta_1\, x_1 + \beta_2\, x_2 = z$$

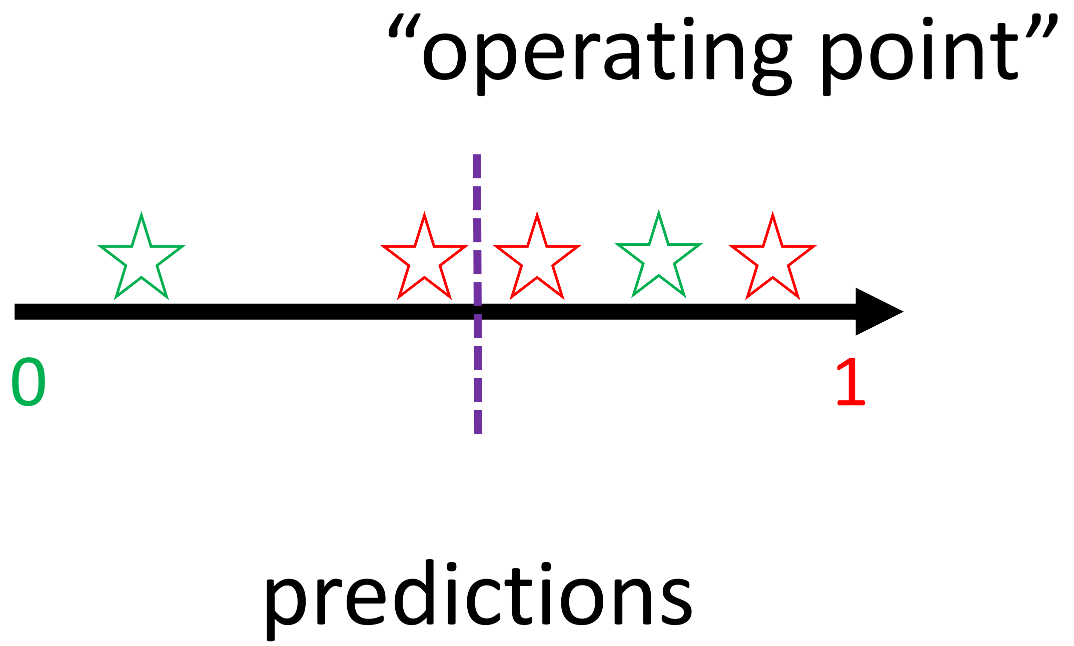
$$\hat{p} = \frac{1}{1+e^{-z}}$$

$$z = 1 * 1 + 3 * 0 + 2 * (-1) = -1$$

$$\hat{p} = \frac{1}{1+e^{-(-1)}} \approx 0.73$$

not a discrete prediction,
needs to be discretized

Discretization of predictions



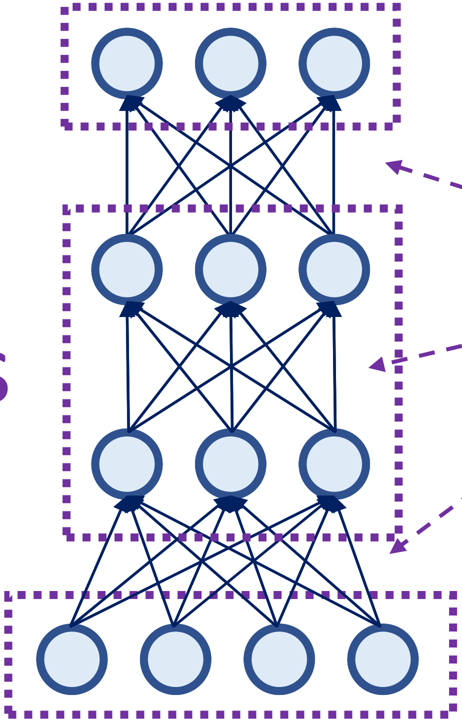
Neural networks

Learning in neural networks: adjusting the weights of the network such that the output matches the training data

output layer

hidden layers

input layer



weights on the edges – learnable parameters

Learning in neural networks

Hyperparameter!

$$\theta_t = \theta_{t-1} - \boxed{\alpha} \frac{\partial L(\theta)}{\partial \theta}\bigg|_{\theta_{t-1}}$$

$$L(\theta) = -\sum_{i=1}^{N} \log \hat{p}_{y_i}(x_i, \theta)$$

Evaluation of classifiers

Two problems:

- $\boxed{\text{How to divide the data set.}}$
- What metrics to use.

How to divide the data set

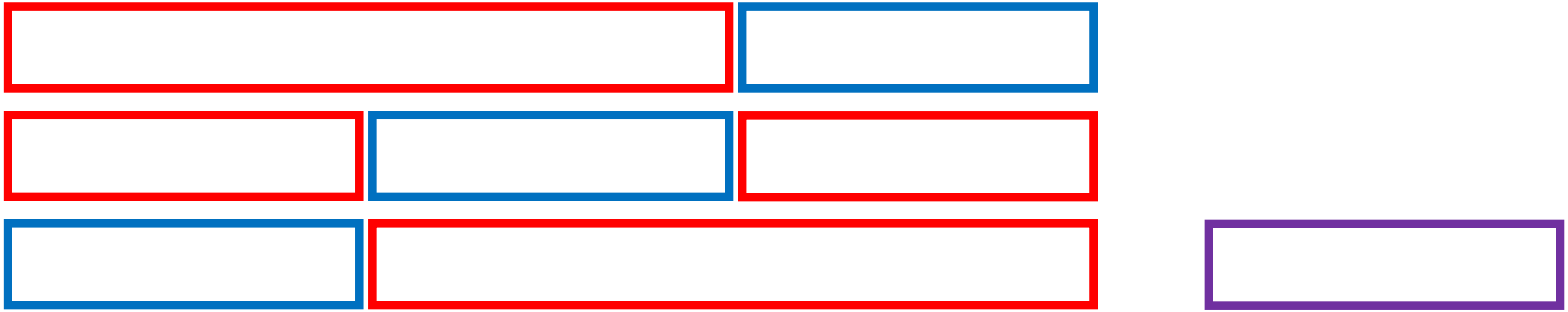
training
(fitting the parameters of the model)

validation
(selecting the hyperparameters)

test
(checking how accurate the model is)

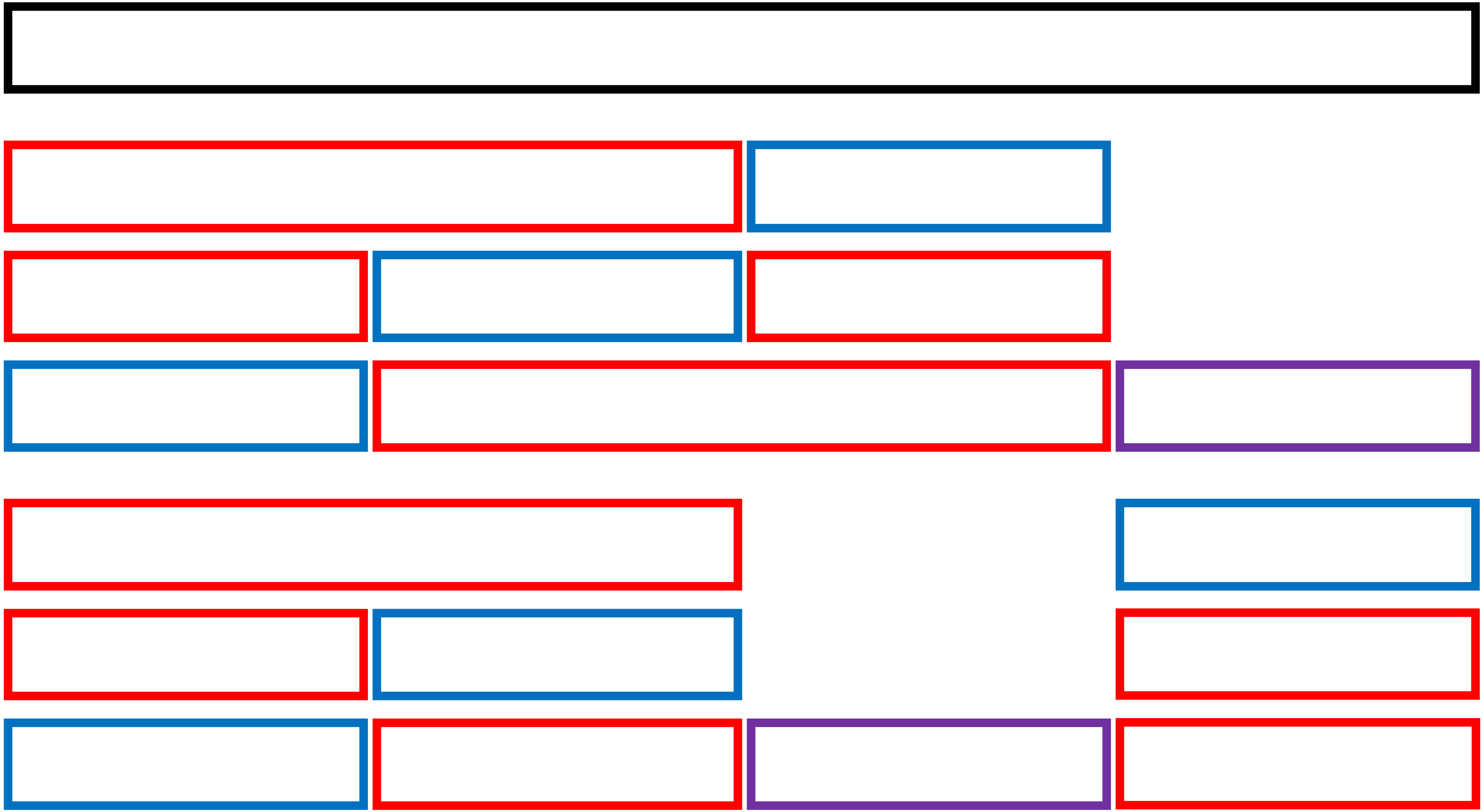
What is the problem with this?

K-fold cross-validation



1. For each set of hyperparameters:
   A. For each $k \in \{1, \dots K\}$.
      a. Train the model on the training set.
      b. Evaluate the model on the validation set.
   B. Select the best model based on the average validation error.
2. Evaluate the best model on the test set.

Nested cross-validation



Cross-validation

- There exists a zoo of cross-validation procedures.
- E.g. Monte Carlo cross-validation, leave-one-out cross-validation, ... You probably won't need them but you should not assume k-fold cross-validation is the only option. Sometimes a different method might be more appropriate.
- K-fold cross-validation is a good procedure for selecting hyperparameters. Its error is not a good estimate of the test error.
- You should evaluate the model on the test set only once.
- Cross-validation is often a bit cumbersome to execute. Use it when you don't have large enough validation and test sets.

Evaluation of classifiers

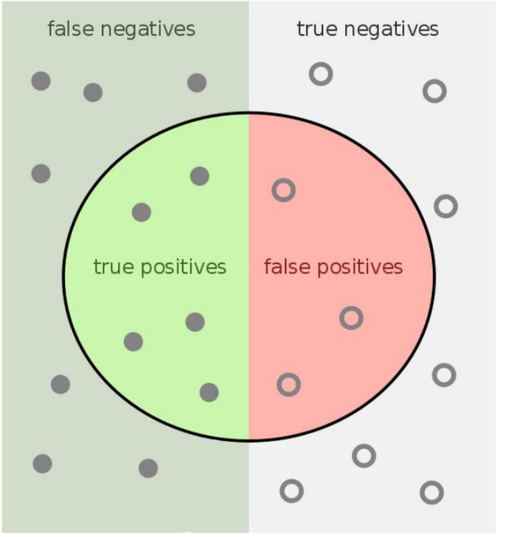
Two problems:

- How to divide the data set.
- What metrics to use.

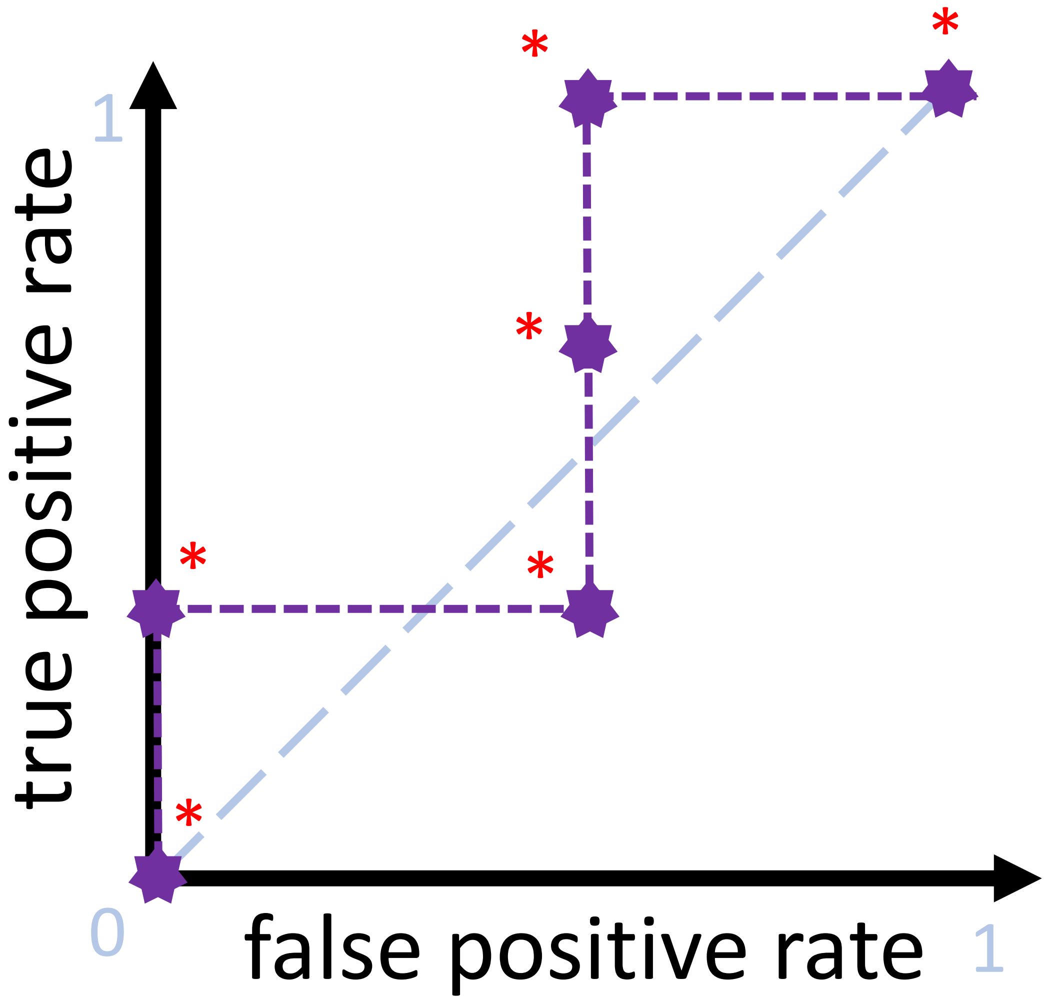
Metrics (for tasks with binary labels)

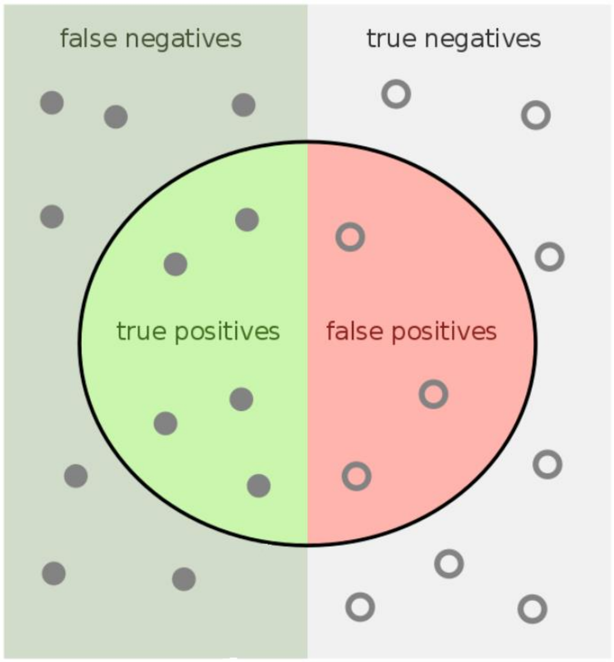
- Accuracy = TP + TN) / everything
- Negative log likelihod
- Precision = TP / (TP + FP)
- Recall = TP / (TP + FN)
- F1 = 2 / (1/precision + 1/recall)
- ROC AUC
- PR AUC

You need to understand what is important for the application you are interested in to select the metric.



Receiver operator characteristic curve (ROC)



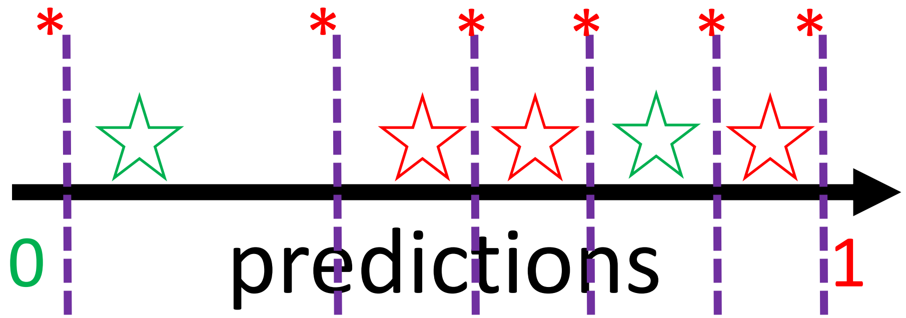


TPR = $TP/(TP+FN)$

es how well

s the two

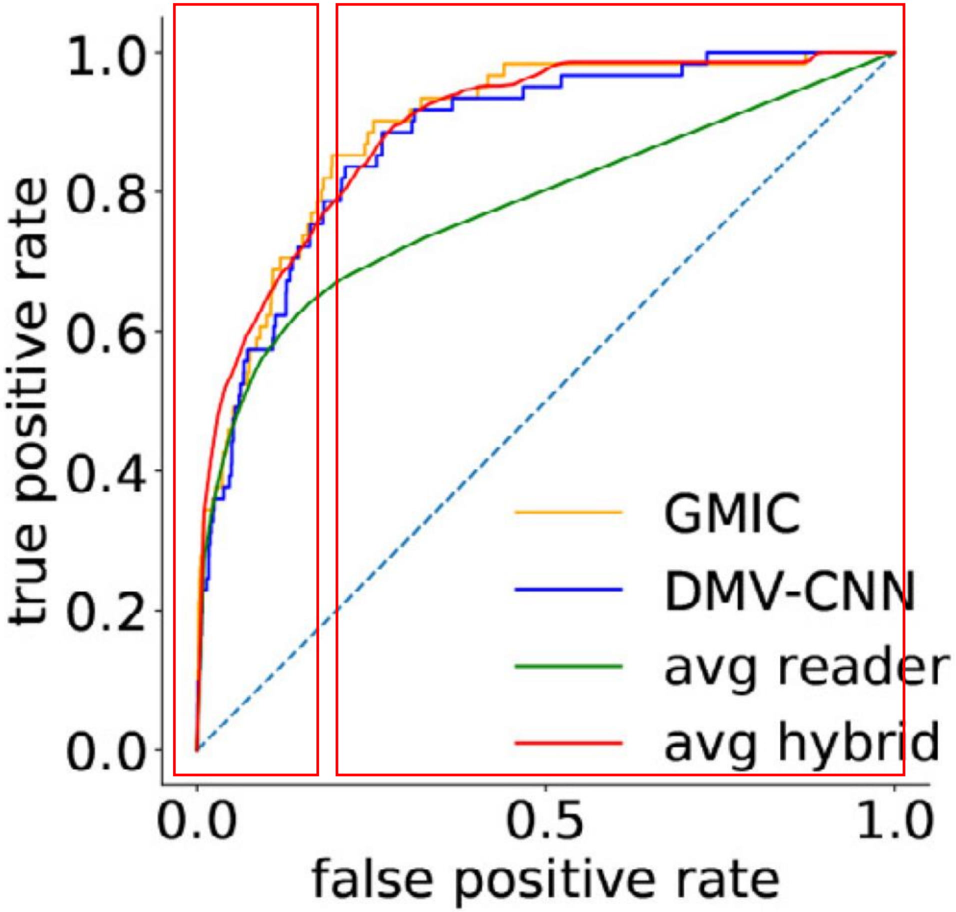
FPR = $FP/(FP+TN)$



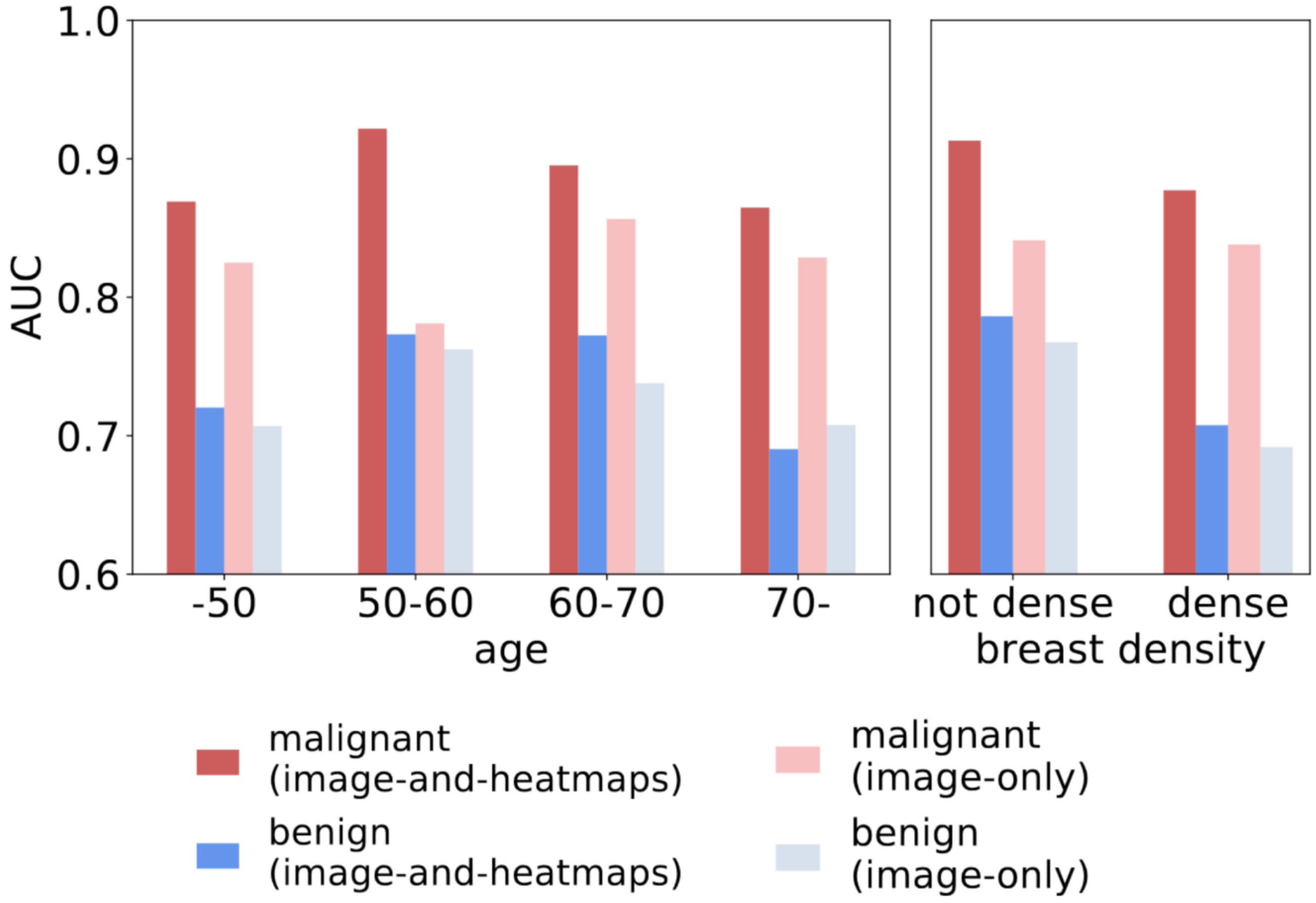
Problems with ROC AUC

AUC doesn't care whether the predictions are calibrated. It only cares about the ordering.

Problems with ROC AUC

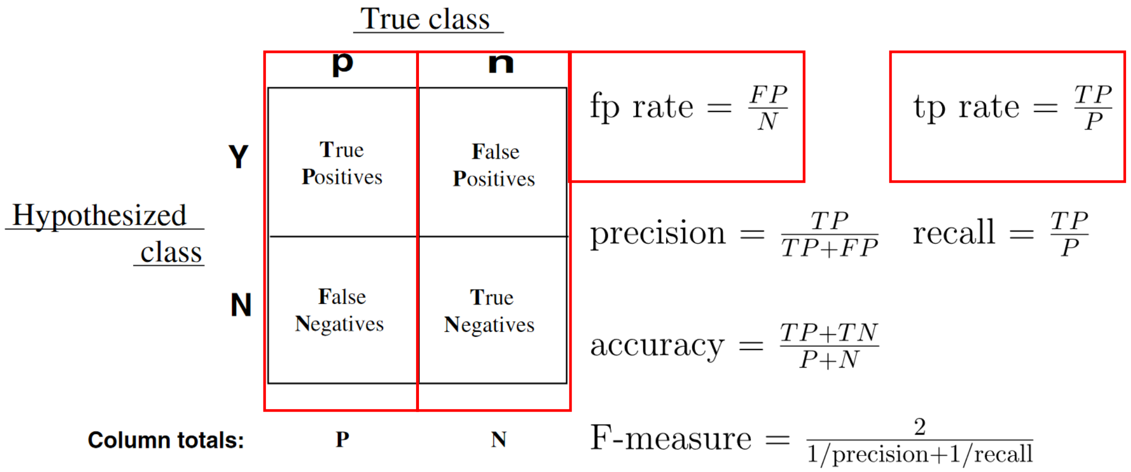


Problems with ROC AUC



Problems with ROC AUC

- How large is its variance with a small test set?
  - There is a *true* AUC which we will never know.
  - We can only estimate it with the test set.
- Surprisingly property: ROC is insensitive to changes in class distribution.



Tom Fawcett. Introduction to ROC analysis. Pattern Recognition Letters, 2007.

Conclusions

- The most important thing is not to mix the training, validation and test sets.
- You have to understand your application very well to understand what metric you should use.
- Evaluation of AI models is kind of hard. Properties of some metrics are somewhat difficult to guess without looking at them mathematically.
- This might be the most important topic in machine learning to understand for you. If you don’t understand it you risk wasting months of work by drawing incorrect conclusions.

Thank you!

kjgeras

k.j.geras@nyu.edu