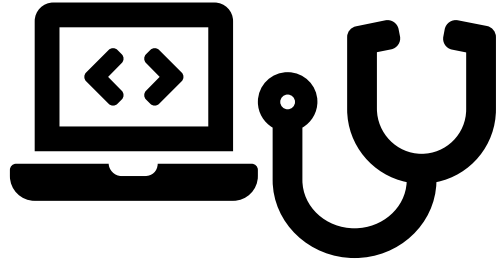
Evaluating your machine learning models beyond the basics

Gaël Varoquaux, Arturo Amor

Inria

Based on Varoquaux & Colliot,
hal.archives-ouvertes.fr/hal-03682454



Outline

1 Performance metrics

Metrics must capture and reflect application

Summary metric: accuracy

Accuracy: counts the fraction of error

Simple summary

Overly simplified picture

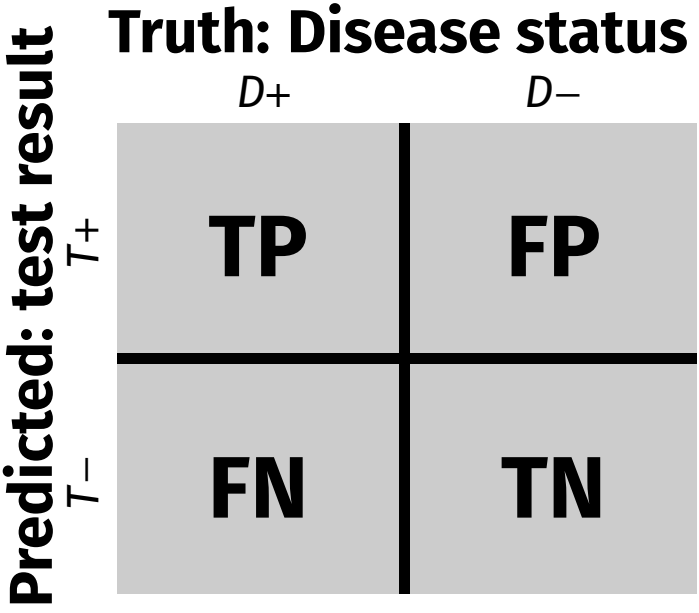
Distinguishing false detections from misses

eg false detection of a brain tumor?

- Very bad if leads to brain surgery
- Minor if leads to confirmatory imaging

Relevant cost (error metric) dependent on application setting

Confusion matrix: beyond accuracy



Metrics on positive and negative classes [Varoquaux and Colliot 2022]

- **Sensitivity (also called recall)**: fraction of positive samples retrieved

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

- **Specificity**: fraction of negative samples actually classified as negative

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

`metrics.recall_score, pos_label=0` for specificity

Metrics on positive and negative classes [Varoquaux and Colliot 2022]

- **Sensitivity (also called recall)**: fraction of positive samples retrieved

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

- **Specificity**: fraction of negative samples actually classified as negative

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

`metrics.recall_score, pos_label=0` for specificity

- **Positive predictive value (PPV, also called precision)**: fraction of the positively classified samples which are indeed positive

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

- **Negative predictive value (NPV)**: fraction of the negatively classified samples which are indeed negative

$$\text{NPV} = \frac{\text{TN}}{\text{TN+FN}}$$

Metrics on positive and negative classes [Varoquaux and Colliot 2022]

T denotes *test*: classifier output; *D* denotes *diseased* status.

- **Sensitivity (also called recall)**: fraction of positive samples retrieved

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP+FN}} \qquad \text{Estimates } \mathcal{P}(T+ \mid D+)$$

- **Specificity**: fraction of negative samples actually classified as negative

$$\text{Specificity} = \frac{\text{TN}}{\text{TN+FP}} \qquad \text{Estimates } \mathcal{P}(T- \mid D-)$$

`metrics.recall_score, pos_label=0` for specificity

- **Positive predictive value (PPV, also called precision)**: fraction of the positively classified samples which are indeed positive

$$\text{PPV} = \frac{\text{TP}}{\text{TP+FP}} \qquad \text{Estimates } \mathcal{P}(D+ \mid T+)$$

- **Negative predictive value (NPV)**: fraction of the negatively classified samples which are indeed negative

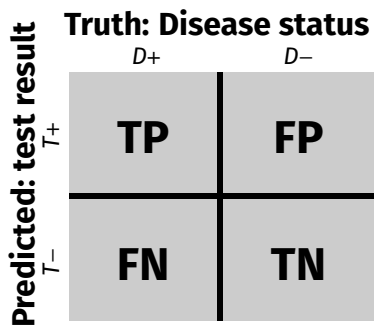
$$\text{NPV} = \frac{\text{TN}}{\text{TN+FN}} \qquad \text{Estimates } \mathcal{P}(D- \mid T-)$$

Performance metrics

From accuracy to balanced accuracy

Accuracy
uninformative under class imbalance

90% of class 0
$\Rightarrow$ predicting only class 0 gives Acc=90%



Balanced accuracy: errors on class 0 and class 1

- Sensitivity (= recall): fraction of class 1 retrieved.
- Specificity: fraction of class 0 actually classified as 0.
- Balanced accuracy: $\frac{1}{2}$ (sensitivity + specificity)

`sklearn.metrics.balanced_accuracy_score`

Asking the right question: $\mathcal{P}(T+|D+)$ vs $\mathcal{P}(D+|T+)$

- Sensitivity: $\mathcal{P}(T+|D+)$ Specificity: $\mathcal{P}(T-|D-)$
- More interesting: $\mathcal{P}(D+|T+)$ Positive predictive value:

(via Bayes' theorem)

Diseased

$$\mathcal{P}(D+ \mid T+) = \frac{\text{sensitivity} \times \text{prevalence}}{(1 - \text{specificity}) \times (1 - \text{prevalence}) + \text{sensitivity} \times \text{prevalence}}.$$

Test positive

Drawback: depends on prevalence

$\Rightarrow$ Characterizes not only the classifier, but also the dataset

Odds ratios for invariance to sampling

Definition: Odds of a $$O(a) = \frac{P(a)}{1 - P(a)}$$

Likelihood ratio of positive class:

$$\text{LR+} = \frac{O(D+|T+)}{O(D+)} = \frac{\text{Sensitivity}}{1 - \text{Specificity}}$$

- Independent of class prevalence[1]
- Use prevalence on target population to compute $O(D+|T+)$

Useful to extrapolate from case-control to target population

In sklearn dev: `sklearn.metrics.class_likelihood_ratios`

[1] Proof in Varoquaux & Colliot "Evaluating machine learning models and their diagnostic value"

Notebook

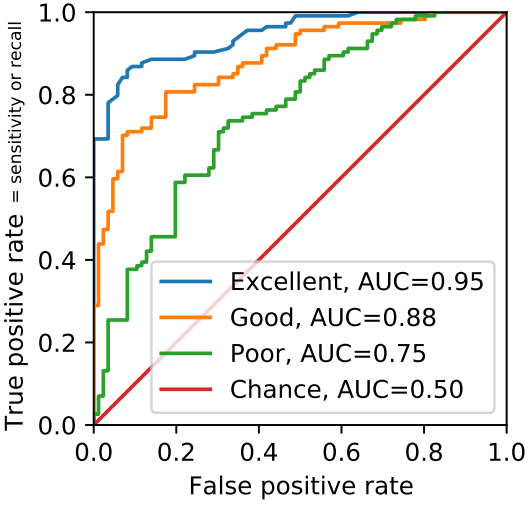
https://github.com/ArturoAmorQ/euroscipy_2022_evaluation

`1_evaluation_tutorial.ipynb`

1 Performance metrics

- Basic classification metrics
- Evaluating imbalanced binary classification
- Multi-threshold metrics for classification
- Confidence scores and calibration

Multi-threshold metrics: ROC curve



Classifiers about continuous scores

ROC obtained by varying the threshold

Summary metric: AUC

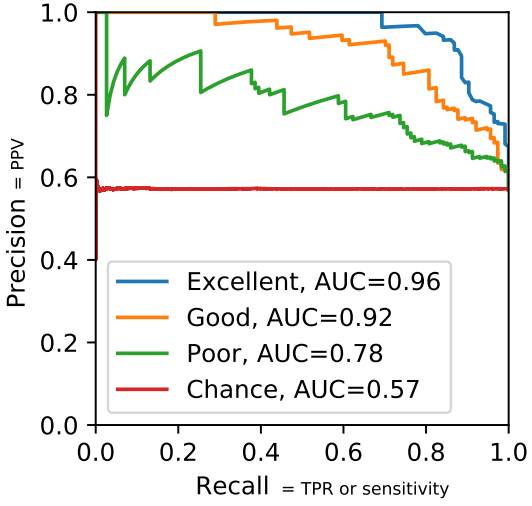
Area Under the Curve

Chance = .5

`sklearn.metrics.roc_auc_score`

`sklearn.metrics.RocCurveDisplay`

Multi-threshold metrics: Precision Recall curve



Better dynamical range than ROC for imbalanced classes

Area Under the Curve:

Chance depends on class imbalance

`sklearn.metrics.average_precision_score`

`sklearn.metrics.PrecisionRecallDisplay`

Notebook

https://github.com/ArturoAmorQ/euroscipy_2022_evaluation

`2_roc_pr_curves_tutorial.ipynb`

Interpreting classifier score as a probability? – Calibration

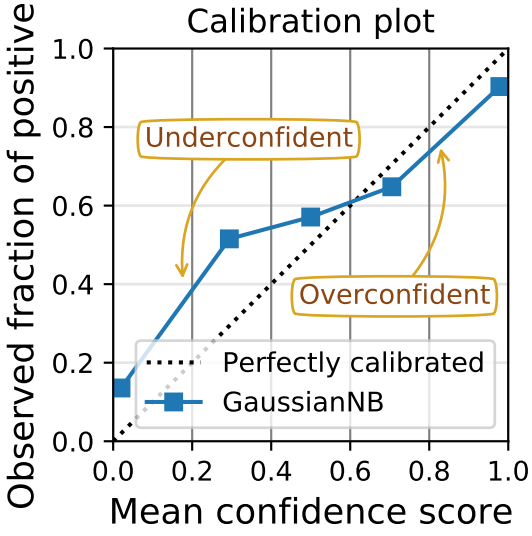
Calibration

Average error rate for all samples with score s is s

Computed in bins on score s

`sklearn.calibration.CalibrationDisplay`

Average error on all bins



⚠ Does not control individual probabilities

Classifier score as a probability? – Individual probabilities

Does the classifier approach $\mathcal{P}(y|X)$?

Observed (binary) label

$$\text{Brier score} = \sum_i (\hat{s}_i - y_i)^2$$

Confidence score

Minimal for $\hat{s} = P(y|X)$

`sklearn.metrics.brier_score_loss`

Choose well your metrics

- Machine learning research chases metrics
  These should reflect application as well as possible

- Worry about $\mathcal{P}(D+|T+)$
  - Accuracy reasonable proxy only for balanced classes
  - LR+ interesting to keep in mind

- Worry about uncertainty
  - Calibration quantifies average errors
  - Brier scores controls individual uncertainty

- A single number does not tell the whole story

2 Evaluation procedures

Controlled estimation of $e = \mathbb{E}_{x,y\sim\mathcal{D}}\left[l(f(x), y)\right]$

Corresponding research paper: [Bouthillier... 2021]

Definitions: what are we benchmarking?

Senario 1: *a prediction rule*:

We are given $f : \mathcal{X} \to \mathcal{Y}$

Evaluating `model.predict`

Senario 2: *a training procedure*:

We are given: a procedure that outputs a prediction rule $\hat{f}$ from training data $(\mathbf{X}, \mathbf{y}) \in (\mathcal{X} \times \mathcal{Y})^n$

Evaluating `model.fit + model.predict`

Definitions: what are we benchmarking?

Senario 1: *a prediction rule*:

We are given $f : \mathcal{X} \to \mathcal{Y}$

Evaluating `model.predict`

For application claims: *eg* medicine

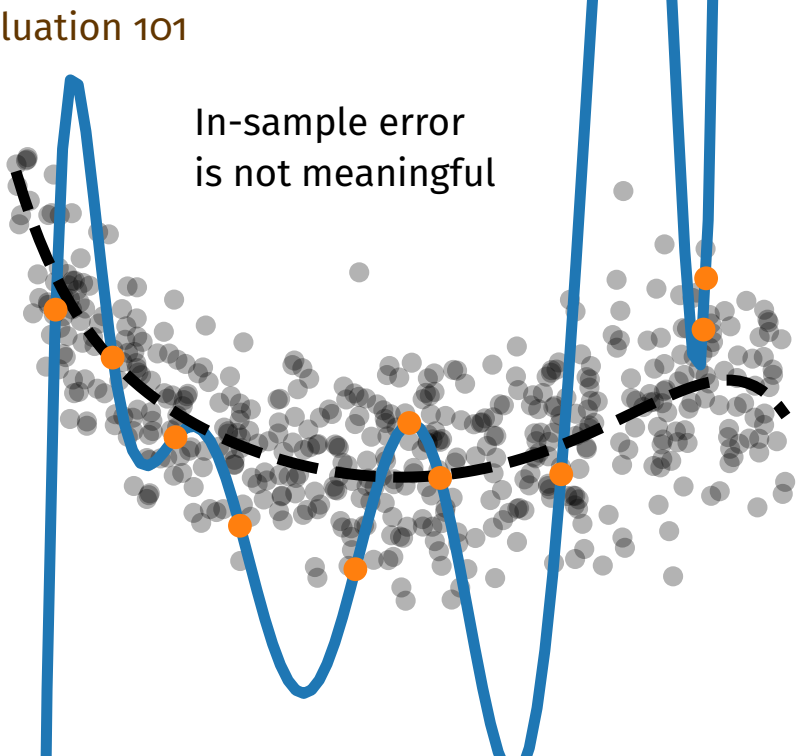
Senario 2: *a training procedure*:

We are given: a procedure that outputs a prediction rule $\hat{f}$ from training data $(\mathbf{X}, \mathbf{y}) \in (\mathcal{X} \times \mathcal{Y})^n$

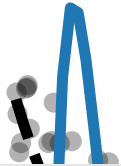
Evaluating `model.fit + model.predict`

For machine-learning research (claims on algorithms)

Model evaluation 101



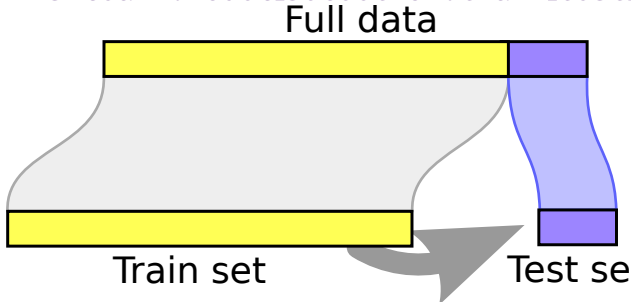
Model evaluation 101



In-sample error
is not meaningful

Typically:

`sklearn.model_selection.train_test_split`



10% test: trade off better learning vs better estimation

2 Evaluation procedures

- Evaluating a prediction rule
- Evaluating a training procedure

External validation

We are given $f : \mathcal{X} \rightarrow \mathcal{Y}$ (fitted model)

X_{test} different enough from X_{train}

- No repeated acquisition of same individual in train & test

[Little... 2017]

- Ideally: show generalization to new site, later in time...

X_{test} representative of target population

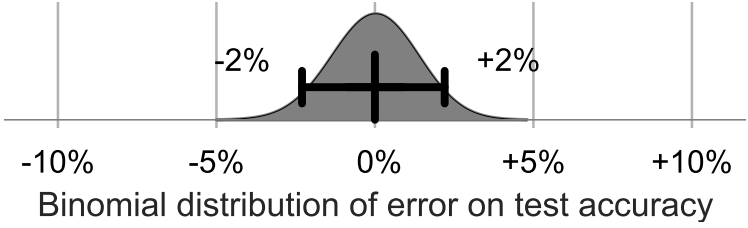
Sample X_{test}:

- to match statistical moments

Evaluation error: Sampling noise on test set

Evaluation quality is limited by number of test examples

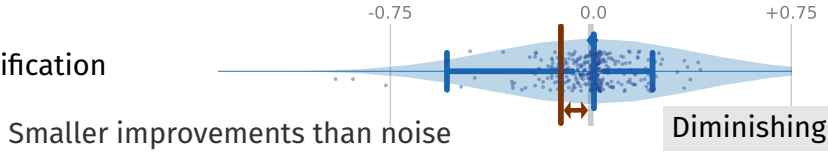
Sampling noise[1] for $n_{\text{test}} = 1000$:



[Varoquaux 2018]

[1] The data at hand (*eg* the test set) is just a small sample of the full population "in the wild", and sampling other data will lead to other results.

Evaluation noise is not negligible – in Kaggle competitions



[Varoquaux and Cheplygina 2022]

Confidence intervals & statistical testing

Evaluate uncertainty to know when to stop, what to trust

(diminishing returns, creeping complexity)

Confidence interval[1]: Range of values compatible with the observations

For accuracy[2]: binomial

N	65%	80%	90%	95%
100	[-9.0% 9.0%]	[-8.0% 8.0%]	[-6.0% 5.0%]	[-5.0% 4.0%]
1000	[-3.0% 2.9%]	[-2.5% 2.4%]	[-1.9% 1.8%]	[-1.4% 1.3%]
10000	[-0.9% 0.9%]	[-0.8% 0.8%]	[-0.6% 0.6%]	[-0.4% 0.4%]
100000	[-0.3% 0.3%]	[-0.2% 0.2%]	[-0.2% 0.2%]	[-0.1% 0.1%]

Table from [Varoquaux and Colliot 2022]

[1] Technically not making the difference with a credible interval

[2] Also for sensitivity, NPV, PPV, see [Varoquaux and Colliot 2022]

2 Evaluation procedures

- Evaluating a prediction rule
- Evaluating a training procedure

Benchmarking to conclude on good training procedures

- We are given: a procedure that outputs a prediction rule $\hat{f}$ from training data $(\mathbf{X}, \mathbf{y}) \in (\mathcal{X} \times \mathcal{Y})^n$

  We want machine-learning research claims
  (choosing one model over another)

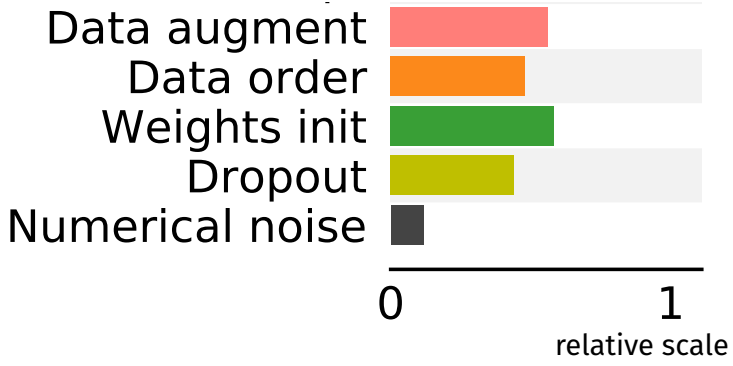
- Many arbitrary components

  `random_state = 3407` ??

Useless to tune random seeds
will not carry over to new training data

Arbitrary variance in a machine-learning benchmark

Variance when rerunning an evaluation,
modifying arbitrary elements:

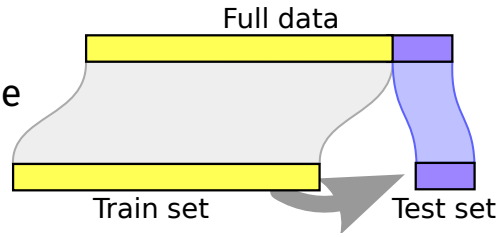


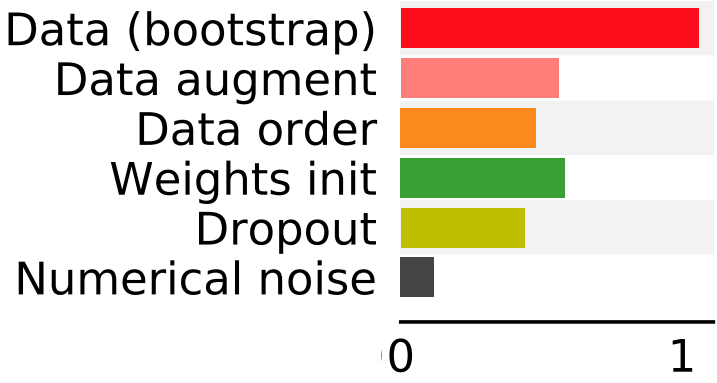
[Bouthillier... 2021]

Uncertainty due to test set sampling

The test set remains a limited sample of the population

The train-test split is an arbitrary choice



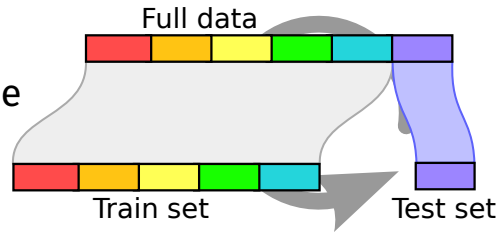


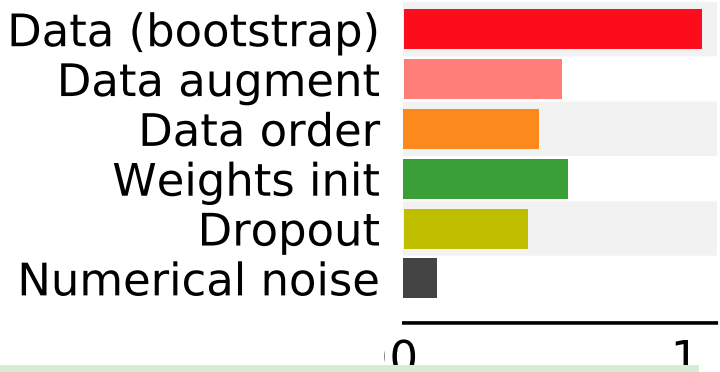
[Bouthillier... 2021]

Uncertainty due to test set sampling

The test set remains a limited sample of the population

The train-test split is an arbitrary choice





Better evaluation

Sample multiple times these arbitrary choices

[Bouthillier... 2021]

Notebook

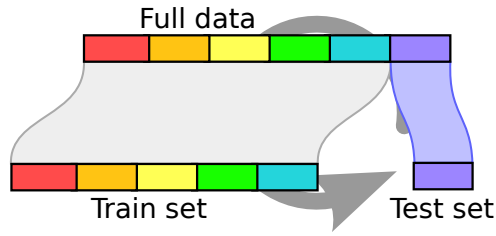
https://github.com/ArturoAmorQ/euroscipy_2022_evaluation

`3_uncertainty_in_metrics_tutorial.ipynb`

Better evaluation procedure

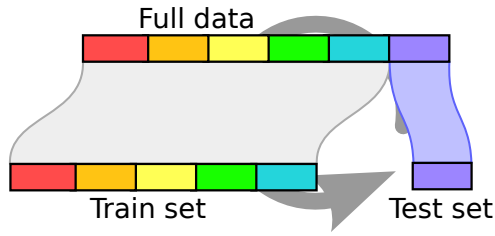
- **Cross-validation**
  multiple train-test splits

- **Randomize all arbitrary factors in folds**



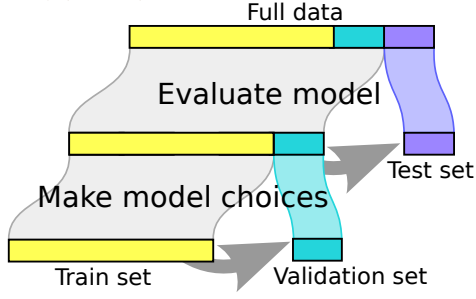
Better evaluation procedure

- Cross-validation
  multiple train-test splits
- Randomize all arbitrary factors in folds



But, a full learning pipeline comes with hyper-parameters

These are set to minimize error on left-out data

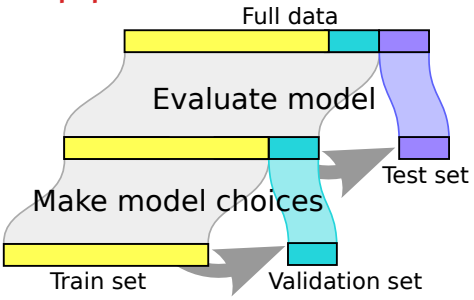


Benchmarks accounting for hyper-parameter selection

Setting hyper-parameters is part of the pipeline
It must be evaluated

- One option:
  train, validation, and test splits
  + manual tuning of hyperparameters

Rampant overfit of validation set
[Makarova... 2021]

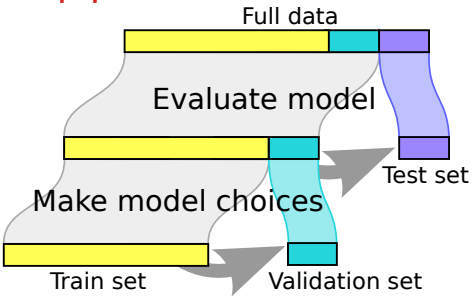


Benchmarks accounting for hyper-parameter selection

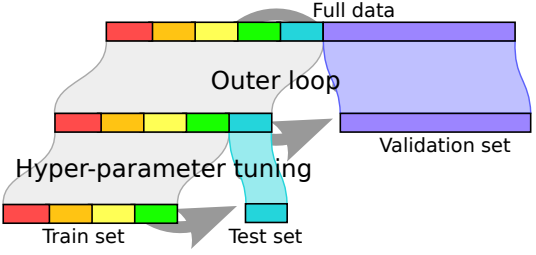
Setting hyper-parameters is part of the pipeline
It must be evaluated

- One option:
  train, validation, and test splits
  + manual tuning of hyperparameters

Rampant overfit of validation set
[Makarova... 2021]



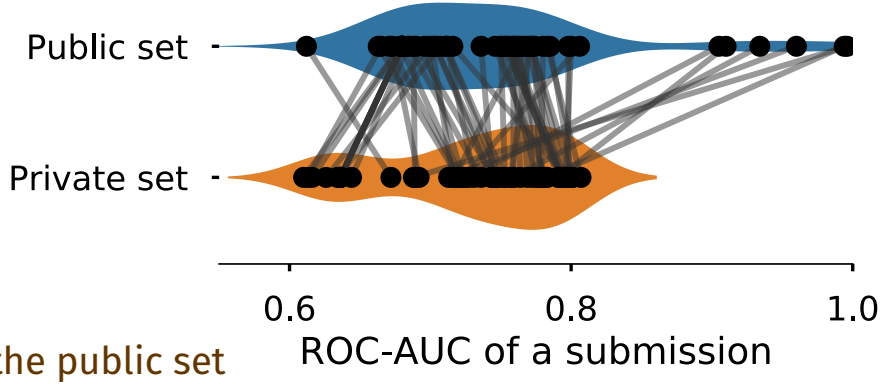
- Multiple splits
  + automated tuning of hyperparameters



Brain imaging prediction

- Autism status prediction
- 10 000 € incentives

[Traut... 2022]



Analysts overfit the public set

Sub-optimal hyper-parameters on baseline models routinely lead to invalid conclusions

See refs in [Bouthillier... 2021]

Hyper-parameter optimization procedures

Grid search

Vary each hyper-parameter on a grid

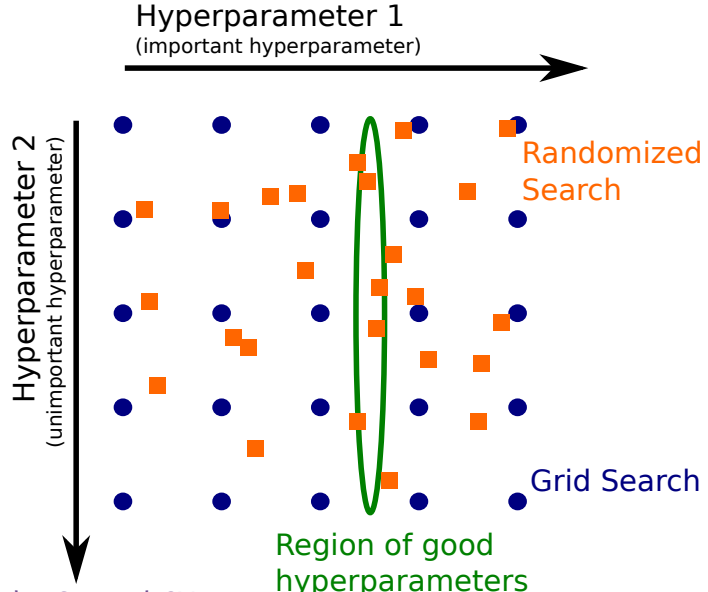
`sklearn.model_selection.GridSearchCV`

Hyper-parameter optimization procedures

- Grid search
- Random search

  [Bergstra and Bengio 2012]

  prefer to grid-search for more than 2 params



`sklearn.model_selection.RandomSearchCV`

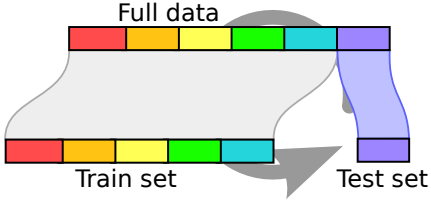
See scikit-learn MOOC, Hyperparameter tuning chapter

Benchmarking training procedures (*eg* to compare them)

Control arbitrary fluctuations (that will not generalize)

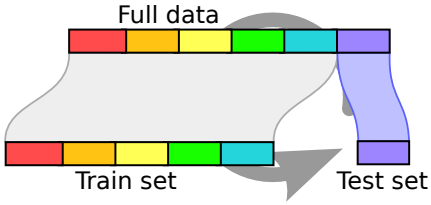
Sample all:

- data sampling
  Multiple train-test splits (cross-validation)
- arbitrary choices (seeds)
  Randomize them all
- hyper-parameters
  Hyper-parameter optimization Expensive to randomize



Accounting for variance in conclusions

Confidence intervals & statistical testing



Statistical testing with multiple folds

Challenge: folds are not independent[1]

⚠ t-test/Wilcoxon across folds are not valid

Don't divide std by number of folds

[1] Train sets overlap, and often test sets also do.

Statistical comparison of pipelines

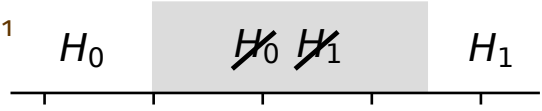
[Bouthillier... 2021]

Hypothesis testing (Neyman-Pearson view)

Two hypothesis, H_0 and H_1

H_1: p_1 outperforms p_2 by a margin[1]

Which is mostly likely?



- Test on $\mathcal{P}(p_1 > p_2) > \delta$ $\delta > .5$: Neyman-Pearson margin
- Evaluate $\mathcal{P}(p_1 > p_2)$ by resampling

  Randomize everything: data splits, seeds,...

Gaussian approximation: amounts to comparing differences to standard deviations

[1] Related to the *rejection region* in the Neyman-Pearson lemma.

Summary – comparing learning procedures

Sample multiple sources of variance:

- data split
- random seeds
- hyper-parameter tuning

Two different cases:

- Concluding on fitted model: a single train-test split
- Concluding on the fitting procedure: many train-test split

Statistical testing: no t-test on cross-validation!

Detect practical differences:

delta in performance > standard deviation

Better experimental procedures

Crack the black box open

A prediction score is seldom insightful

- Ablation studies: remove/change atomic elements
- Learning curves

Better benchmarking in these

- Tune hyper-parameters to the same quality
- Randomize everything
- Account for variance in conclusions

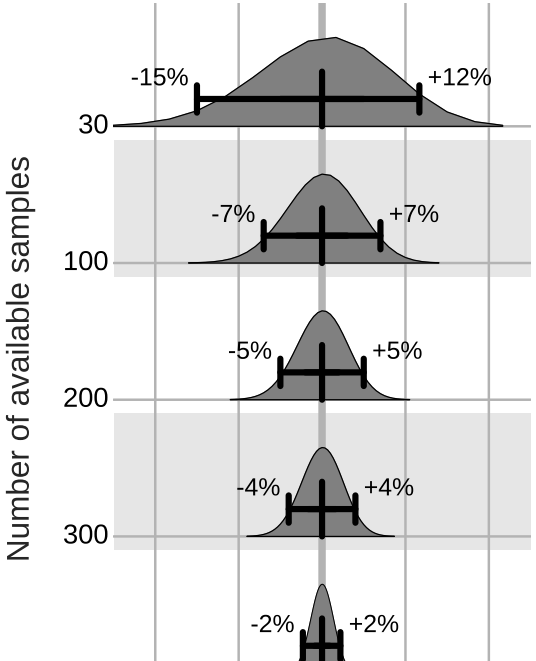
Summary – Better benchmarking procedures

Reminder: Your validation measure is intrinsically unreliable (sampling noise)

An arbitrary choice (random seed) may give seemingly-good results that do not generalize

Optimizing test accuracy will explore the tails

Evaluation is a bottleneck



Model evaluation

Choice of performance metrics

- Should be suited to the application setting
- Machine learning does metric chasing 😬

Evaluation procedures

- Account for variance
- Different sources of variance for applying prediction rules vs learning them

Careful benchmarking is crucial

- Optimistic flukes will not generalize
- What is our purpose? External validity ✊


@GaelVaroquaux

References I

J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281, 2012.

X. Bouthillier, P. Delaunay, M. Bronzi, A. Trofimov, B. Nichyporuk, J. Szeto, N. Mohammadi Sepahvand, E. Raff, K. Madan, V. Voleti, ... Accounting for variance in machine learning benchmarks. *Proceedings of Machine Learning and Systems*, 3, 2021.

M. A. Little, G. Varoquaux, S. Saeb, L. Lonini, A. Jayaraman, D. C. Mohr, and K. P. Kording. Using and understanding cross-validation strategies. perspectives on saeb et al. *GigaScience*, 6(5):1–6, 2017.

A. Makarova, H. Shen, V. Perrone, A. Klein, J. B. Faddoul, A. Krause, M. Seeger, and C. Archambeau. Overfitting in bayesian optimization: an empirical study and early-stopping solution. *arXiv preprint arXiv:2104.08166*, 2021.

N. Traut, K. Heuer, G. Lemaître, A. Beggiato, D. Germanaud, M. Elmaleh, A. Bethegnies, L. Bonnasse-Gahot, W. Cai, S. Chambon, ... Insights from an autism imaging biomarker challenge: promises and threats to biomarker discovery. *NeuroImage*, 255:119171, 2022.

G. Varoquaux. Cross-validation failure: small sample sizes lead to large error bars. *Neuroimage*, 180:68–77, 2018.

References II

G. Varoquaux and V. Cheplygina. Machine learning for medical imaging: methodological failures and recommendations for the future. *NPJ digital medicine*, 5(1):1–8, 2022.

G. Varoquaux and O. Colliot. Evaluating machine learning models and their diagnostic value. https://hal.archives-ouvertes.fr/hal-03682454/, 2022.