Model selection

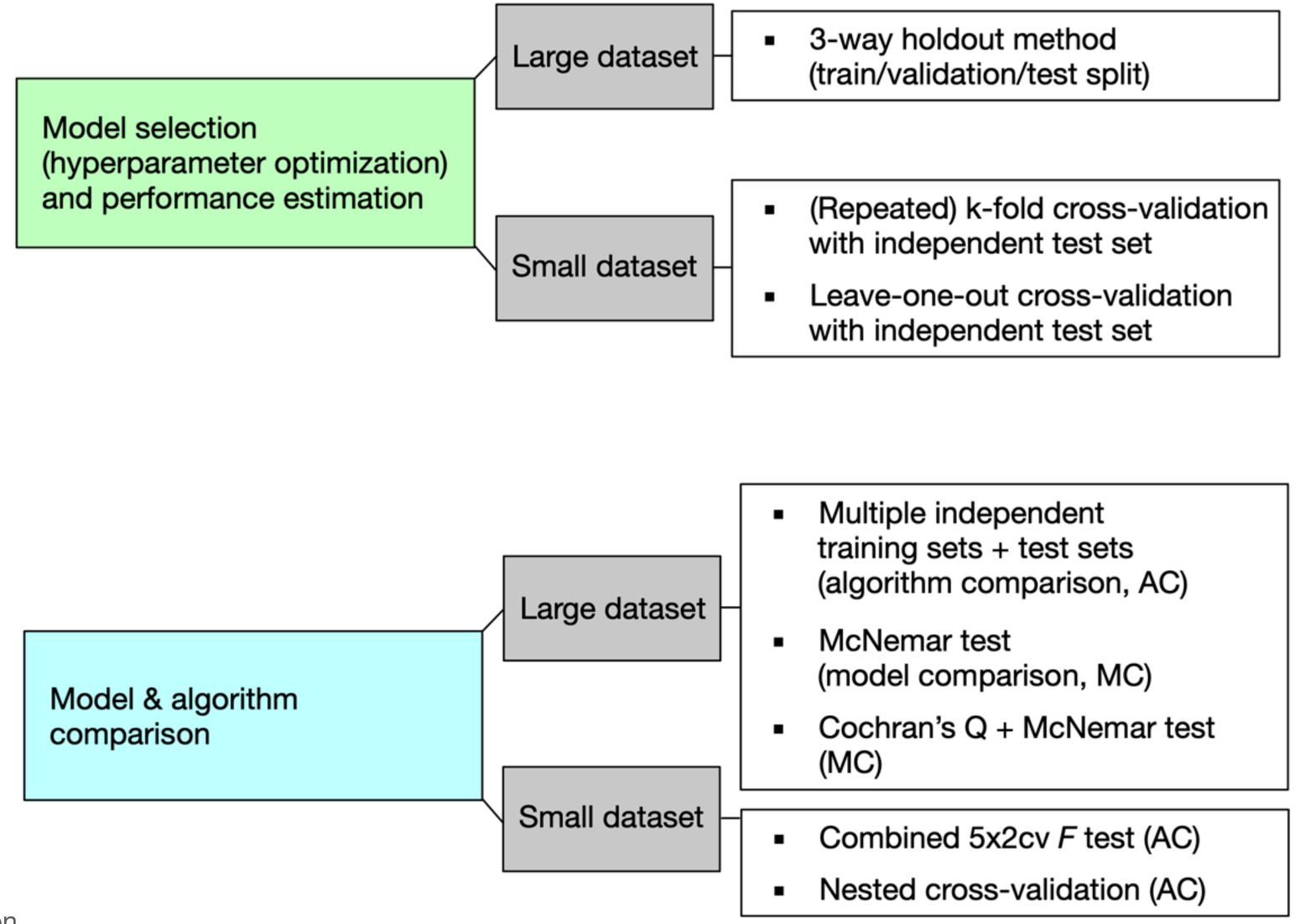
Jason G. Fleischer, Ph.D.
Asst. Teaching Professor
Department of Cognitive Science, UC San Diego

jfleischer@ucsd.edu

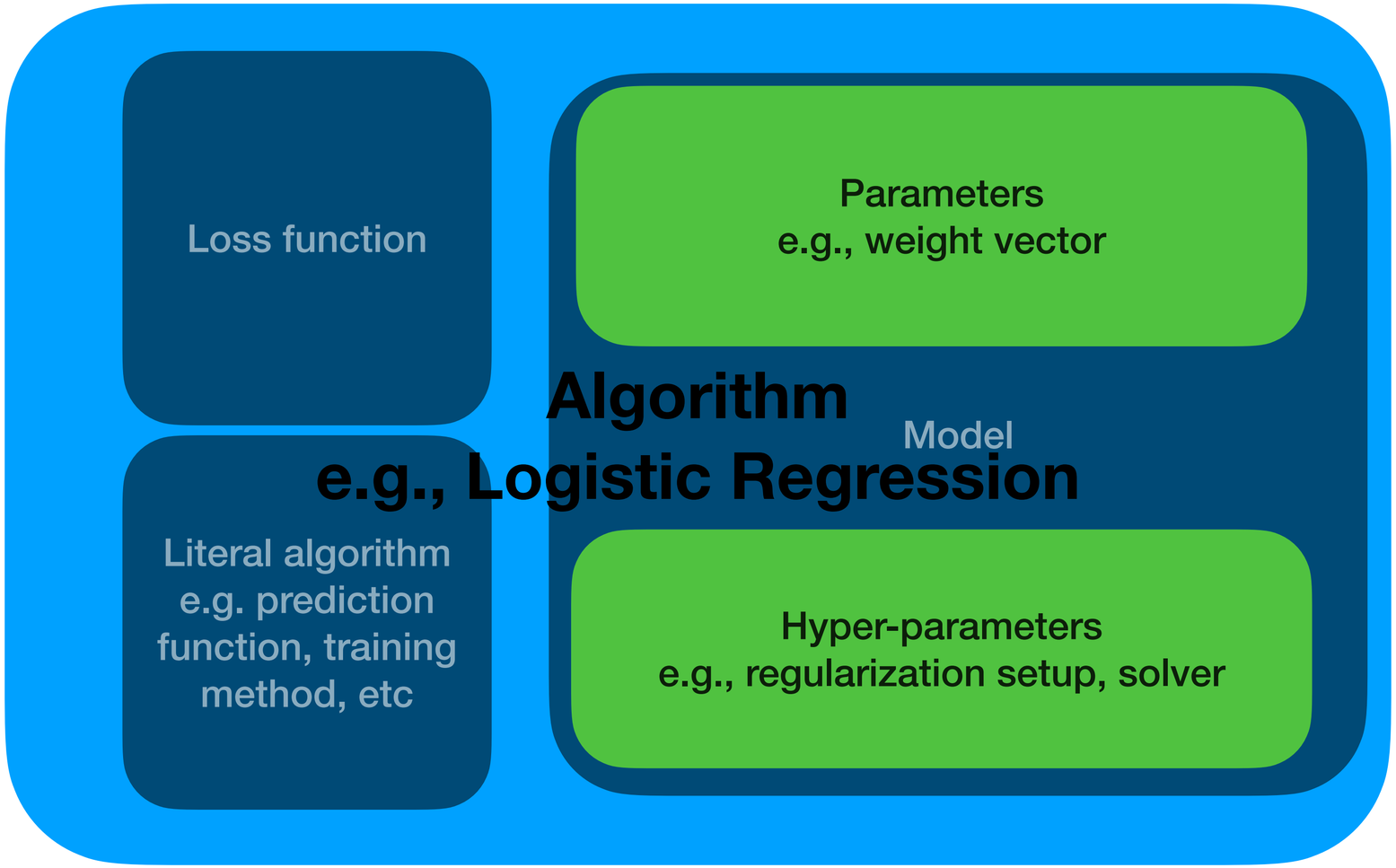
@jasongfleischer

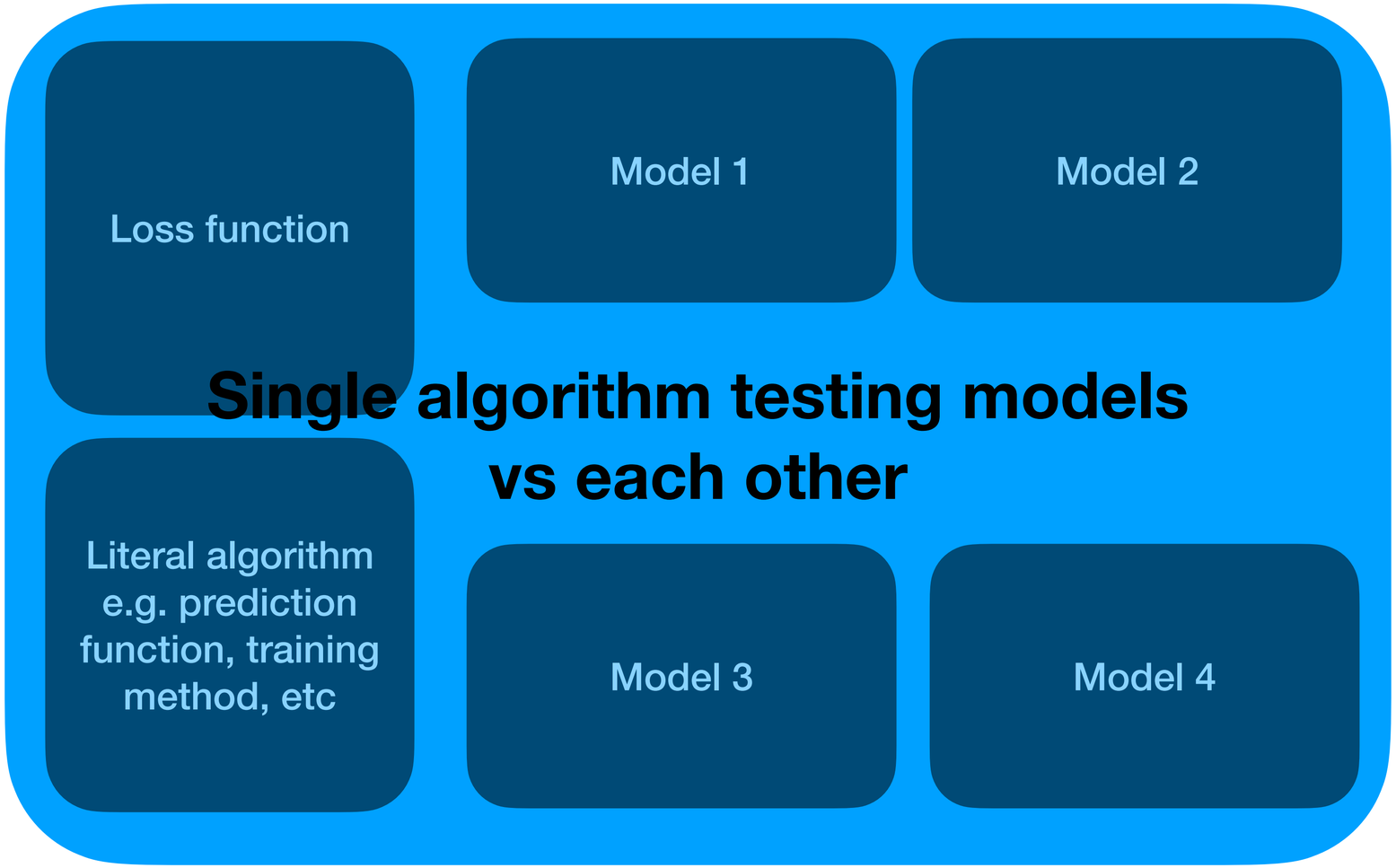
https://jgfleischer.com

Slides in this presentation are from material kindly provided by Sebastian Rashka

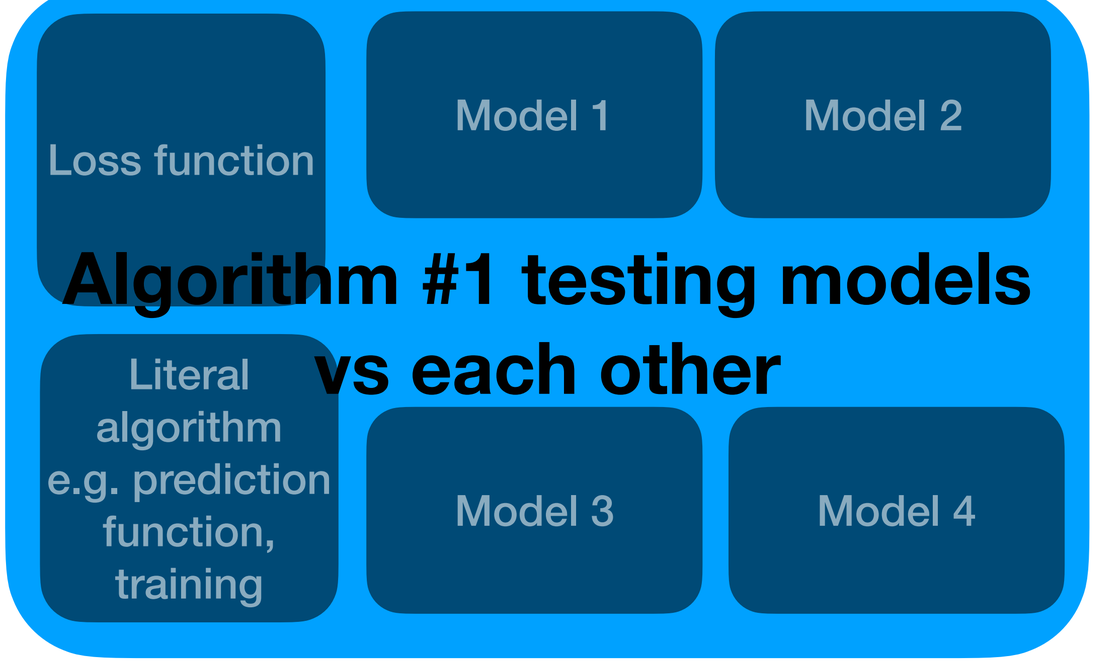


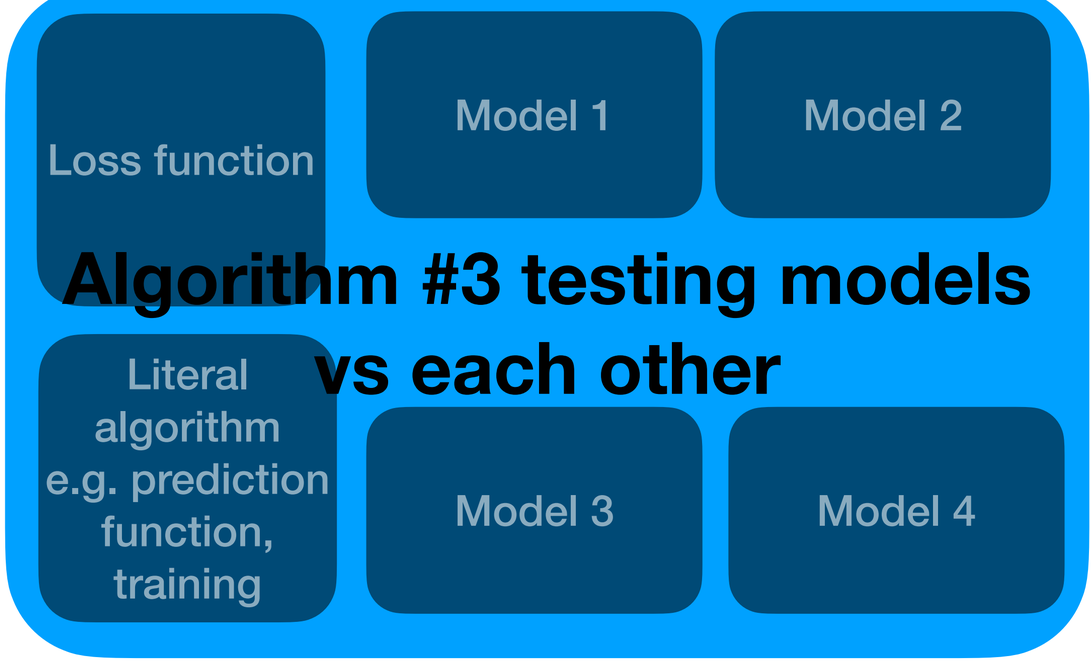
Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning
Sebatian Rashkha
https://arxiv.org/pdf/1811.12808.pdf


Loss function
Parameters
e.g., weight vector
Algorithm
Model
e.g., Logistic Regression
Literal algorithm
e.g. prediction
function, training
method, etc
Hyper-parameters
e.g., regularization setup, solver

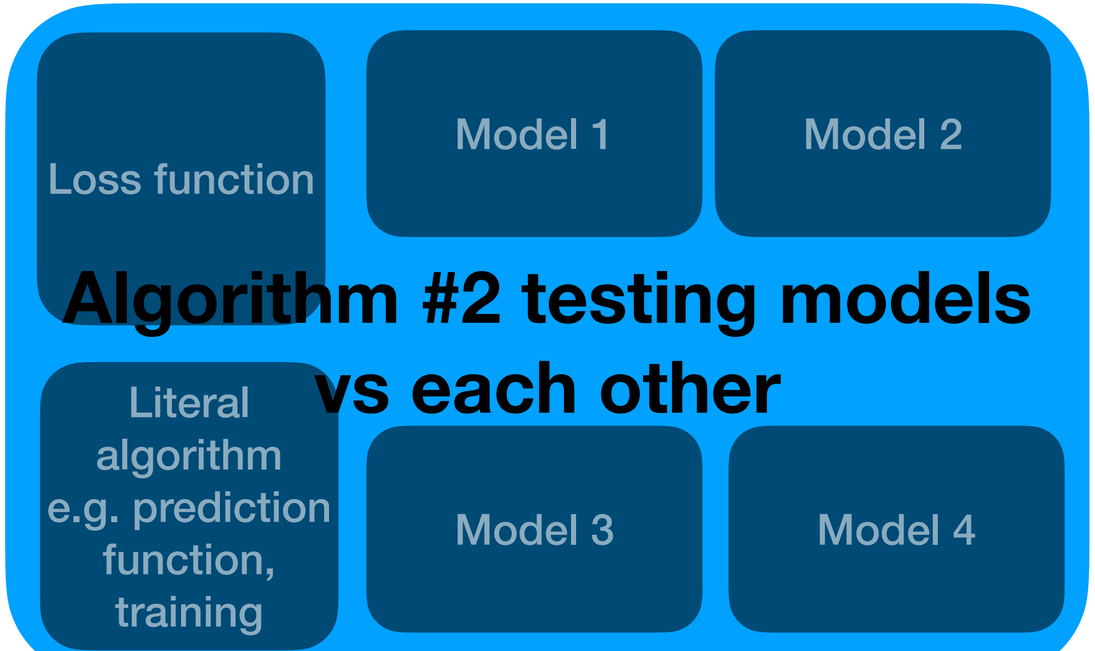


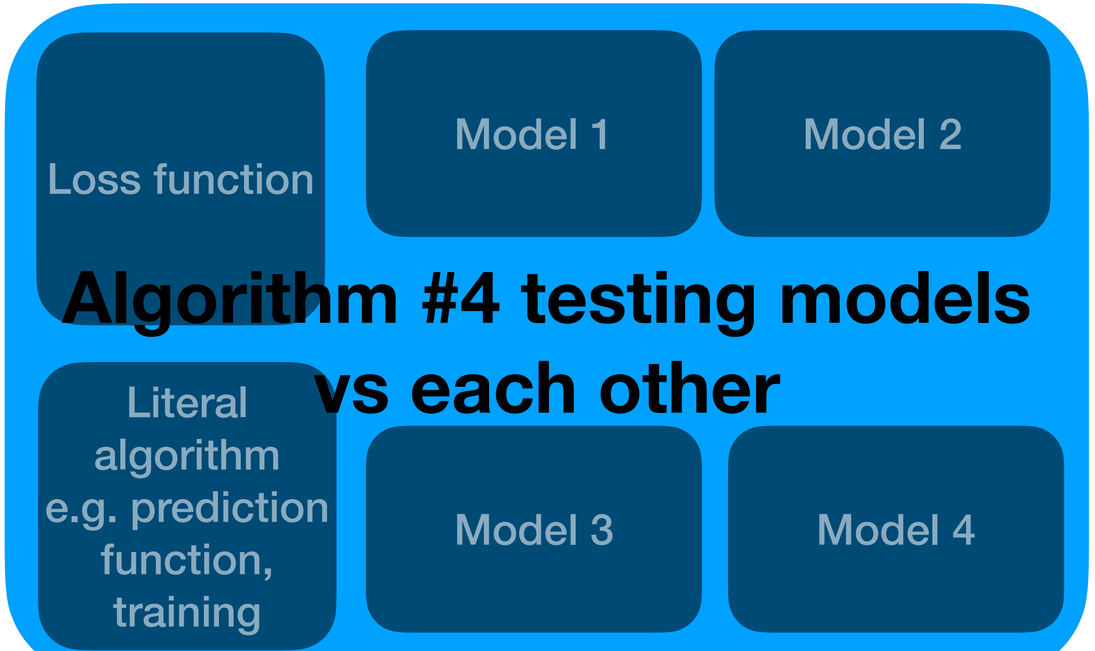
“Model selection”





“Algorithm selection”





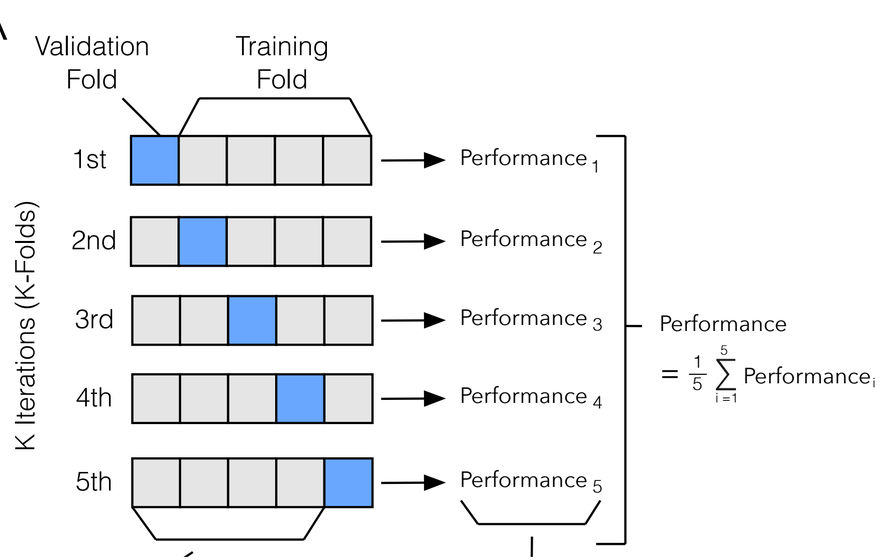
Model or Algorithm selection

Method #0 - Internet sized datasets

- Split data into train, validate, test
- [OPTIONAL] Outer loop… do this T times:
  - For each model in the hyper-parameter space or each algorithm-model combination:
    - Train it on training
    - Predict on the validation set
- Pick the best model or algorithm based on its performance on [the mean across trials] of the validation set
- Train the best version on the whole of training set + validation set
- Test it on the test set to estimate its ability to generalize

METHOD 1 - with enough data to have a good test set

k-Fold Cross-Validation

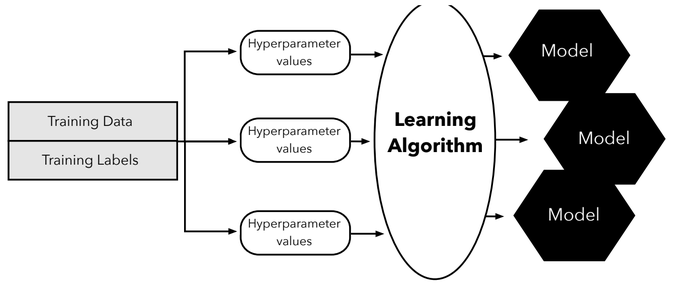


Let's say you had around 8k samples in a dataset

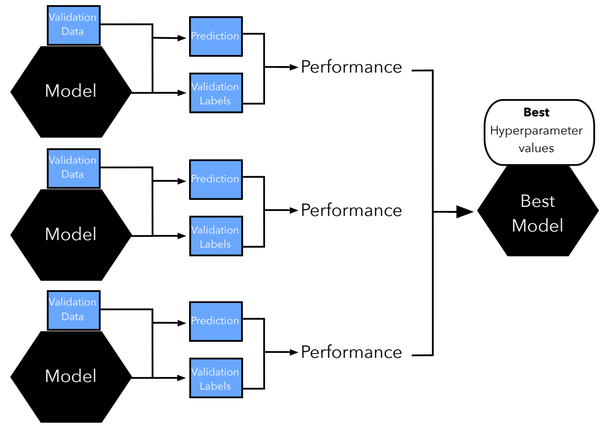
For each trial:

- training set ~ sample 5k (with or w/o) replacement from entire dataset
- Grid search of hyper parameters using 5-fold cross validation on the training set
- Select best model from grid, train on entire training set
- Evaluate best model on the test set (everything not sampled for training)

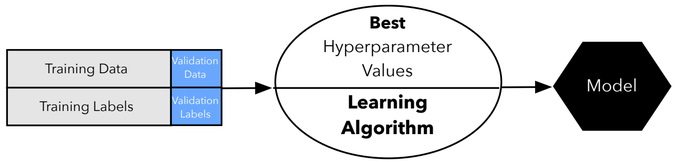
2



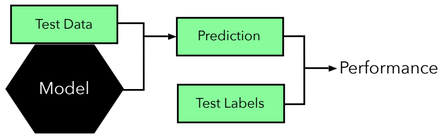
3



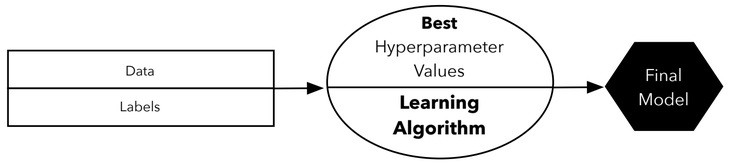
4



5



6



Figures from Sebastian Rashka

METHOD 2 - make the most of a small amount of data

Nested Cross-validation for Algorithm Comparison ONLY!

You've got only ~2000 samples, which is barely enough to fit the data well let alone test

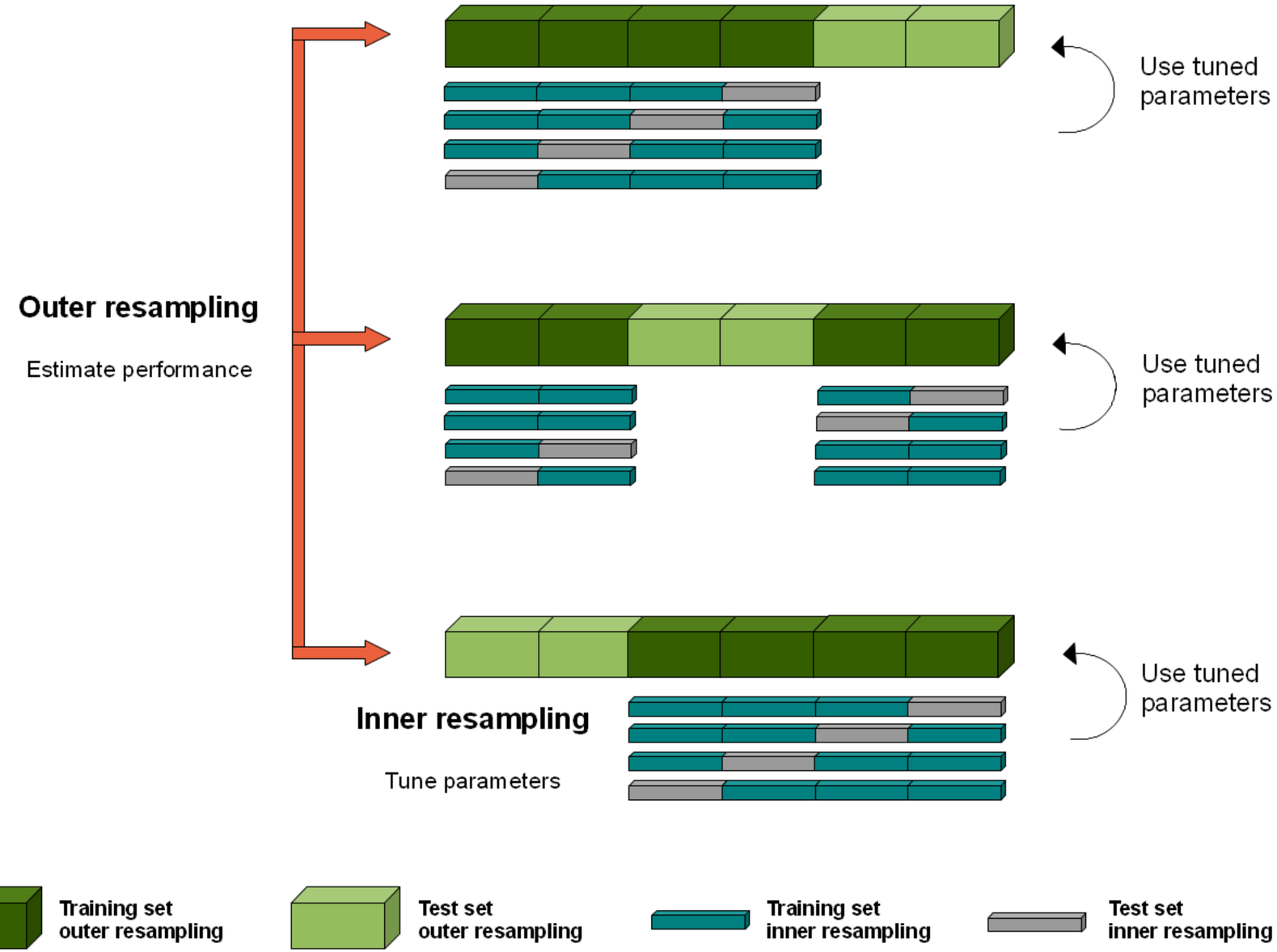
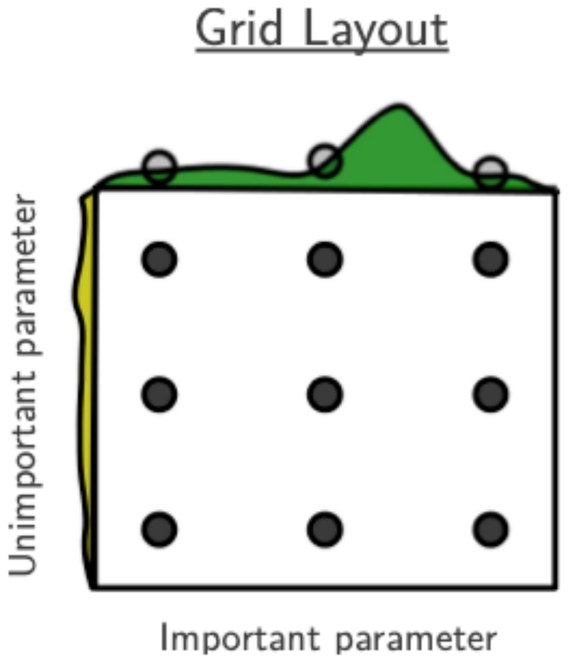


Figure from mlr R package docs

But how do you organize your search of the hyper parameter space?

Grid Search

- Exhaustive search
- Thorough but expensive
- Specify grid for parameter search
- Can be run in parallel
- Can suffer from poor coverage
- Often run with multiple resolutions



Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, *13*(1), 281-305.

Randomized Search

- Search based on a time budget
- Preferred if there are many hyperparameters (e.g. > 3 distinct ones)
- specify distribution for parameter search
- can be run in parallel

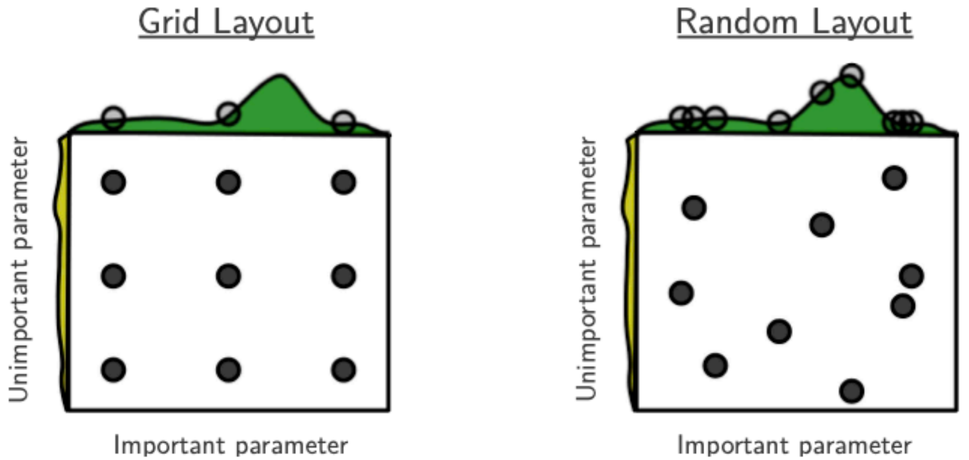
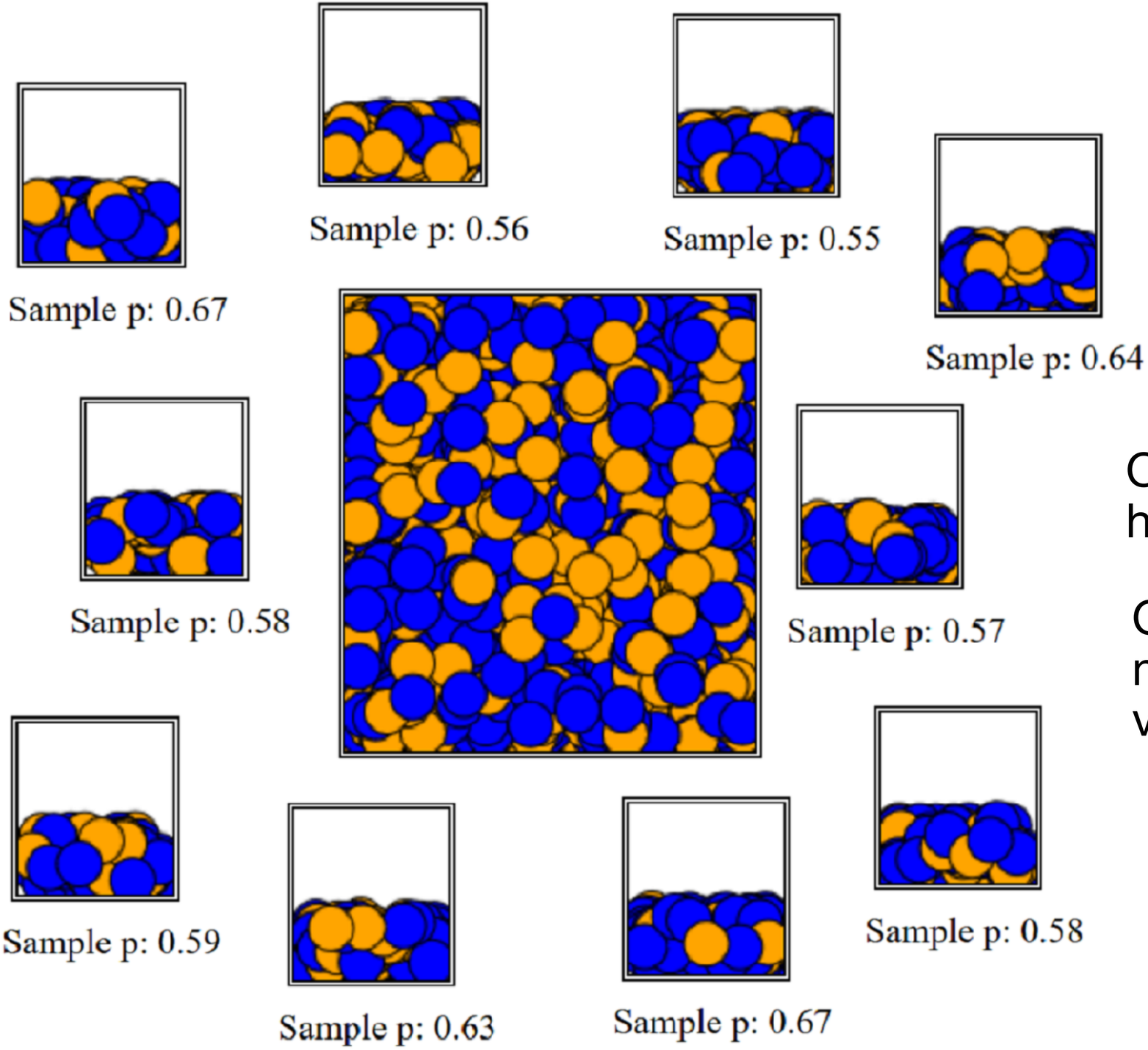


Figure 1: Grid and random search of nine trials for optimizing a function $f(x,y) = g(x) + h(y) \approx g(x)$ with low effective dimensionality. Above each square $g(x)$ is shown in green, and left of each square $h(y)$ is shown in yellow. With grid search, nine trials only test $g(x)$ in three distinct places. With random search, all nine trials explore distinct values of g. This failure of grid search is the rule rather than the exception in high dimensional hyper-parameter optimization.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, *13*(1), 281-305.

Statistical testing

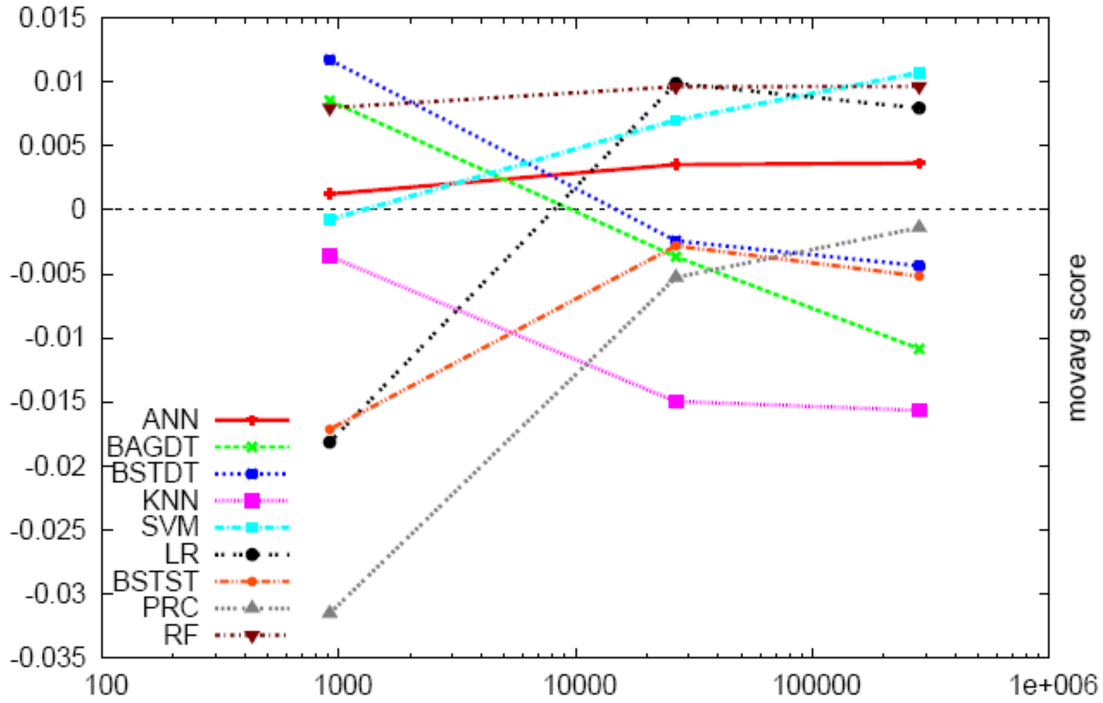
https://sebastianraschka.com/pdf/lecture-notes/stat479fs18/11_eval-algo_notes.pdf



One of the hold out folds

Or one trial of model selection via cross-val

Is any algorithm better than any other?



Caruana et al., ICML 2008

http://icml2008.cs.helsinki.fi/papers/632.pdf

Moving average standardized scores of each learning algorithm as a function of the dimension.

11 binary classification problems whose dimensionality ranges from 761 to 685569

Algorithms that perform well across many different dimensionalities:

(1) random forest (2) neural nets (3) boosted tree (4) SVMs

Frequentist correlated t-test

The correlated t-test is based on the modified Student's t-statistic:

$$t(\boldsymbol{x}, \mu) = \frac{\overline{x} - \mu}{\sqrt{\hat{\sigma}^2(\frac{1}{n} + \frac{\rho}{1-\rho})}} = \frac{\overline{x} - \mu}{\sqrt{\hat{\sigma}^2(\frac{1}{n} + \frac{n_{te}}{n_{tr}})}}, \tag{1}$$

where $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and $\hat{\sigma} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2}$ are the sample mean and sample standard deviation of the data $\boldsymbol{x}$, ρ is the correlation between the observations and μ is the value of the mean we aim at testing. The statistic follows a Student distribution with $n-1$ degrees of freedom:

$$St\left(\bar{x}; n-1, \mu, \left(\frac{1}{n} + \frac{\rho}{1-\rho}\right)\hat{\sigma}^2\right). \tag{2}$$

For $\rho = 0$, we obtain the traditional t-test. For $\rho = \frac{n_{te}}{n_{tot}}$, we obtain the correlated t-test proposed by Nadeau and Bengio (2003) to account for the correlation due to the overlapping training sets. Usually the test is run in a two-sided fashion. Its hypotheses are: $H_0 : \mu = 0; \quad H_1 : \mu \neq 0$. The p-value of the statistic under the null hypotheses is:

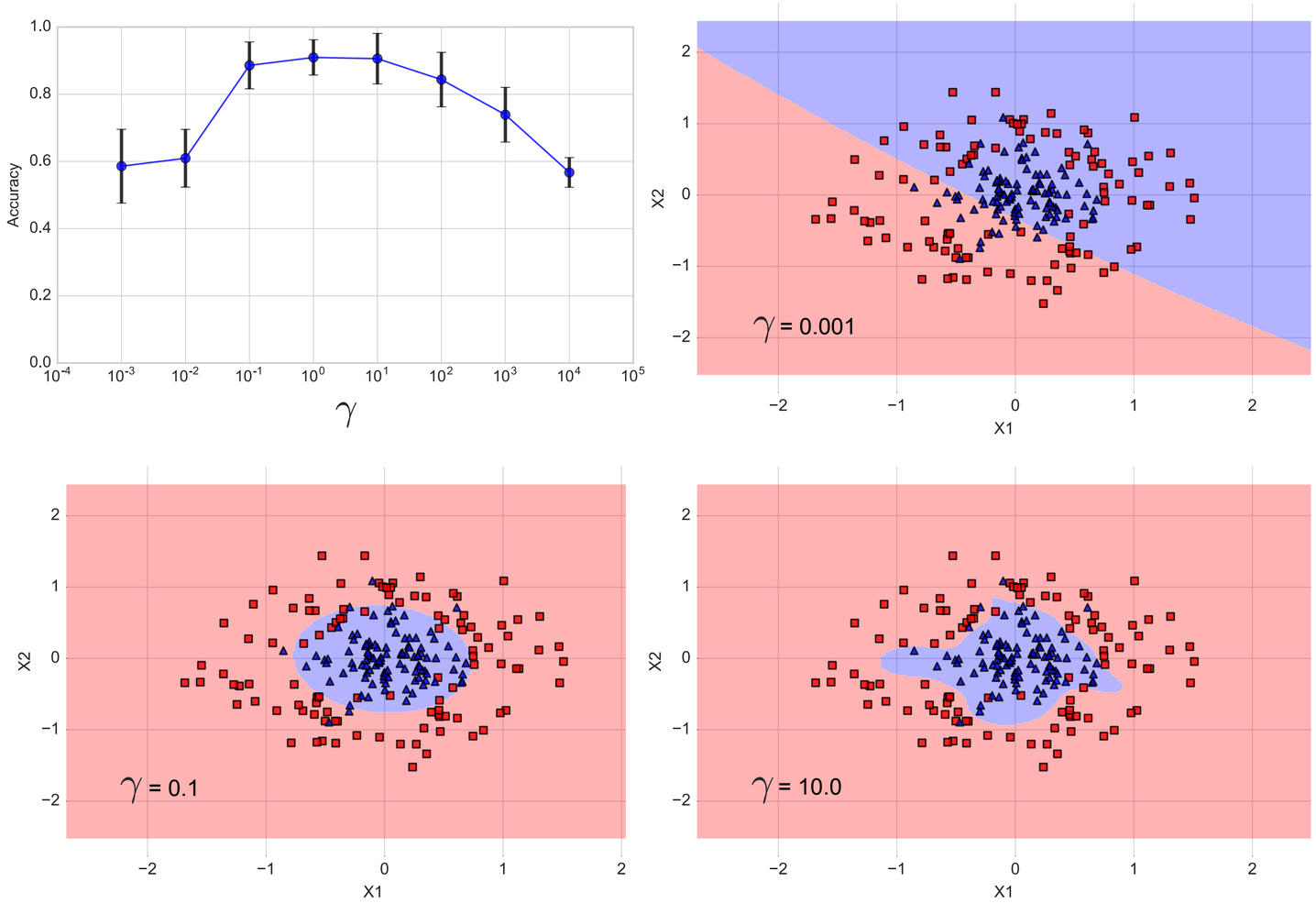
$$p = 2 \cdot (1 - \mathcal{T}_{n-1}(|t(\boldsymbol{x}, 0)|)), \tag{3}$$

where $\mathcal{T}_{n-1}(|t(\boldsymbol{x}, 0)|)$ denotes the cumulative distribution of the standardized Student distribution with $n-1$ degrees of freedom in $|t(\boldsymbol{x}, \mu)|$ for $\mu = 0$. For instance, for the first data set in Table 1 we have that $\bar{x} = -0.0194$, $\hat{\sigma} = 0.01583$, $\rho = 1/10$, $n = 100$ and so $t(\bar{x}, 0) = -3.52$. Hence, the two-sided p-value is $p = 2 \cdot (1 - \mathcal{T}_{n-1}(|t(\boldsymbol{x}, 0)|)) = 0.00065 \approx 0.001$. Sometimes the directional one-sided test is performed. If the alternative hypothesis is the positive one, the hypotheses of the one-sided test are: $H_0 : \mu \leq 0; \quad H_1 : \mu > 0$. The p-value is $p = 1 - \mathcal{T}_{n-1}(t(\boldsymbol{x}, 0))$.

Parsimony Principle

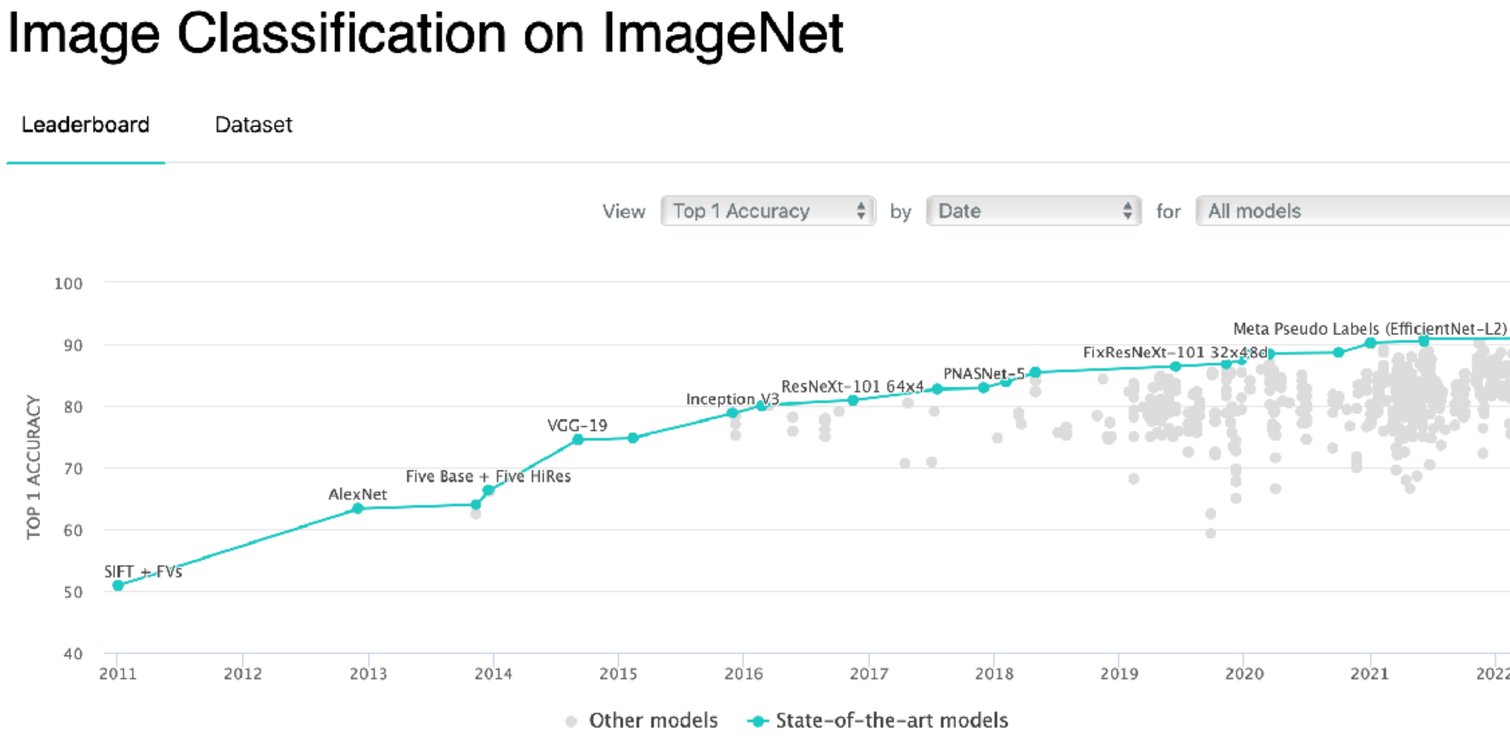
Choose the simplest w/in 1 std error of optimal

Which parameter would you select?



Statistical testing on model performance

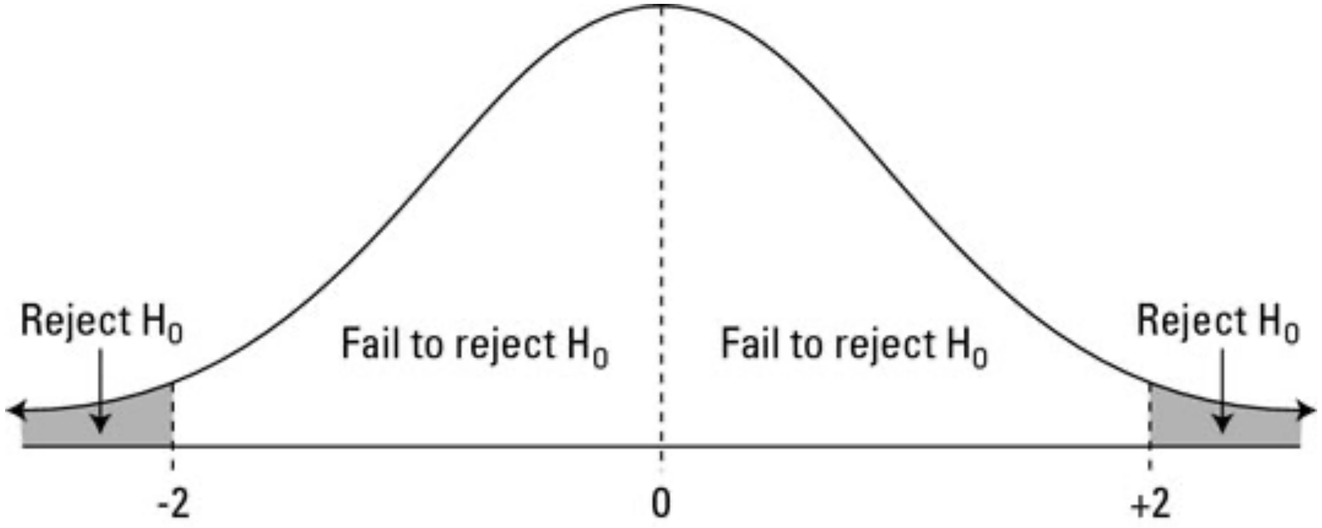
- Testing is almost always paired (over folds of cross validation)
- Distinguish between tests appropriate for algorithm comparison vs model selection (hyperparameter settings)
- Distinguish between test that are computationally efficient vs those that are not
- Distinguish between pair-wise and group-wise tests



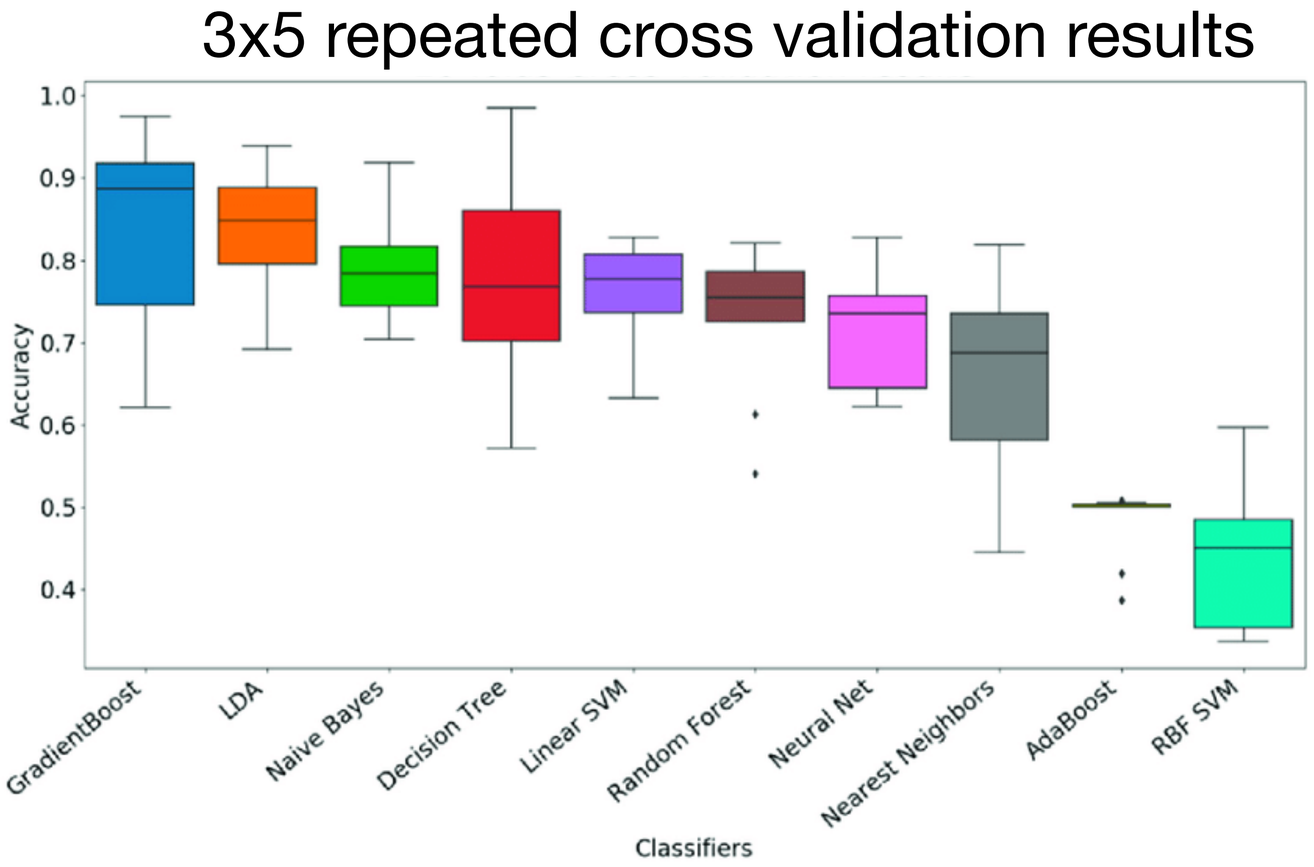
Jason gets grumpy about blindly following methods you don’t understand fully

The p-value

- In range 0,1
- Smaller is support for alternative hypothesis
- Larger is inconclusive
- Ignores effect size!!!@!!! Is the difference practically important?
- Assumes conditions on data
- $P(H|D) = \frac{P(D|H)P(H)}{P(D)}$

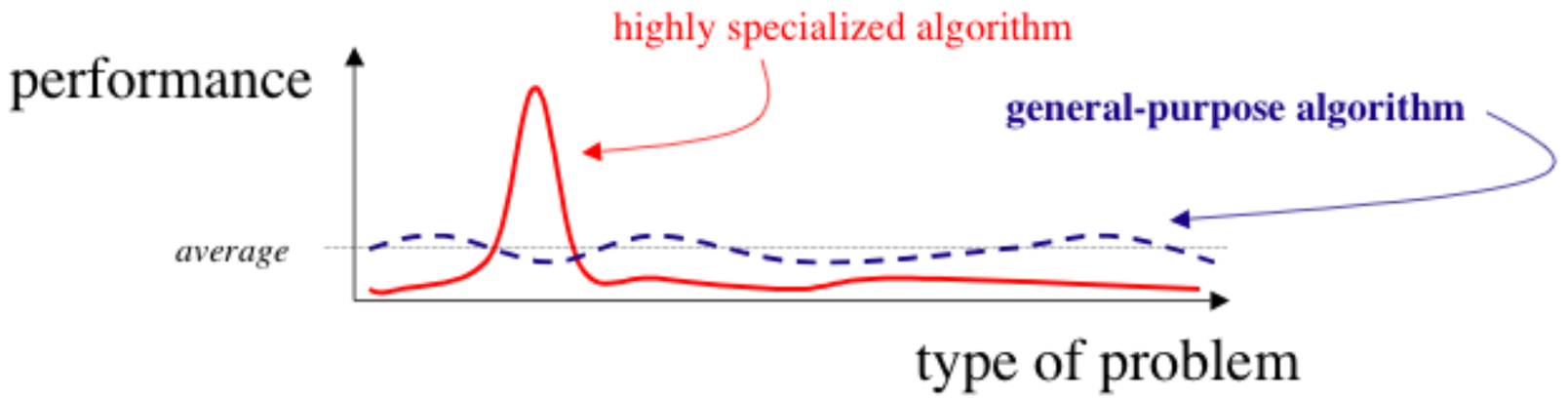


Maybe you don’t need a statistical test



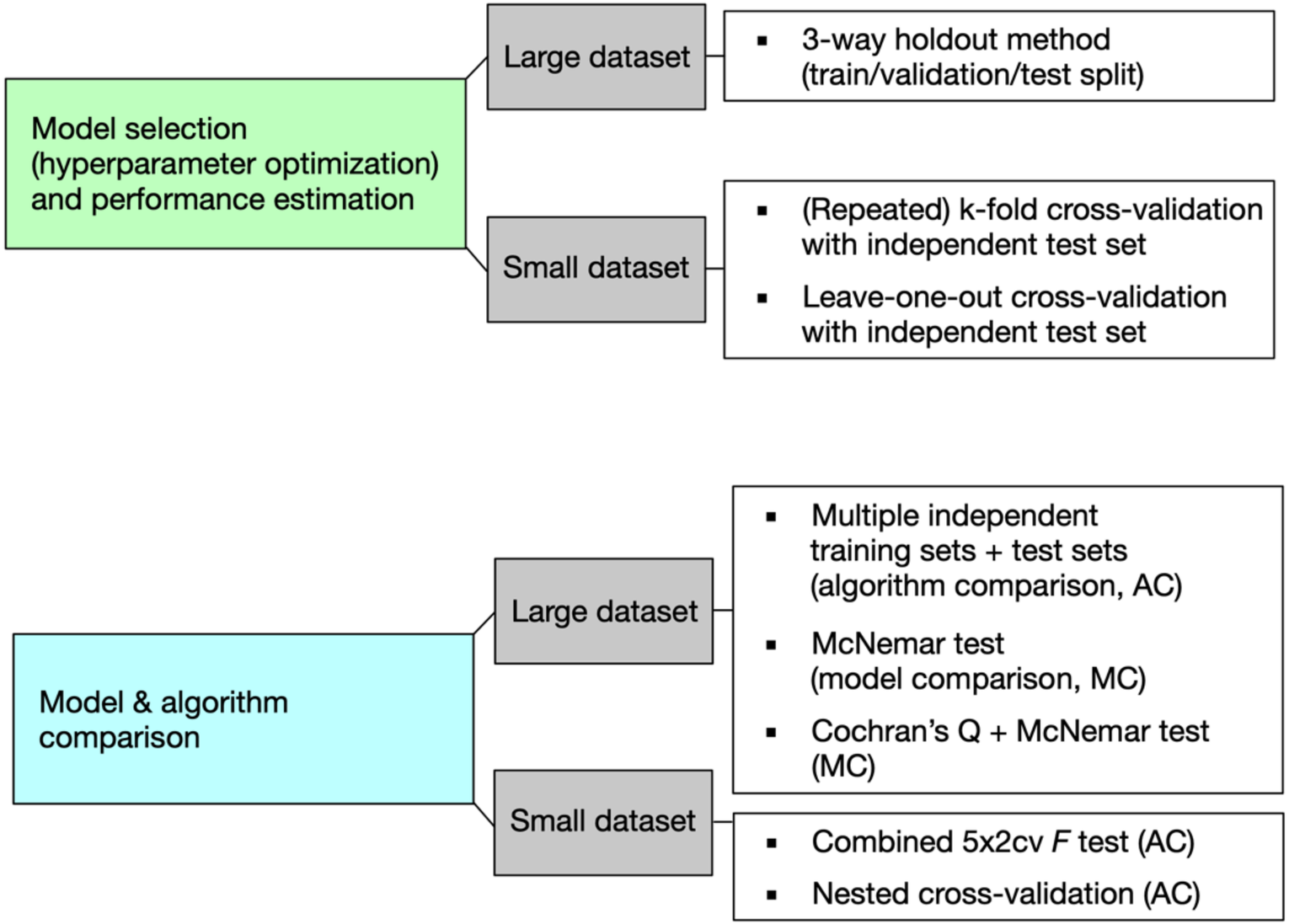
No free lunch theorem

Why even bother??



Wolpert, D. H.; Macready, W. G. (1995). "No Free Lunch Theorems for Search" (PDF). *Technical Report SFI-TR-95-02-010*. Santa Fe Institute.

http://www.no-free-lunch.org


Model selection (hyperparameter optimization) and performance estimation
Large dataset
3-way holdout method (train/validation/test split)
Small dataset
(Repeated) k-fold cross-validation with independent test set
Leave-one-out cross-validation with independent test set
Model & algorithm comparison
Large dataset
Multiple independent training sets + test sets (algorithm comparison, AC)
McNemar test (model comparison, MC)
Cochran's Q + McNemar test (MC)
Small dataset
Combined 5x2cv F test (AC)
Nested cross-validation (AC)