Lecture 11

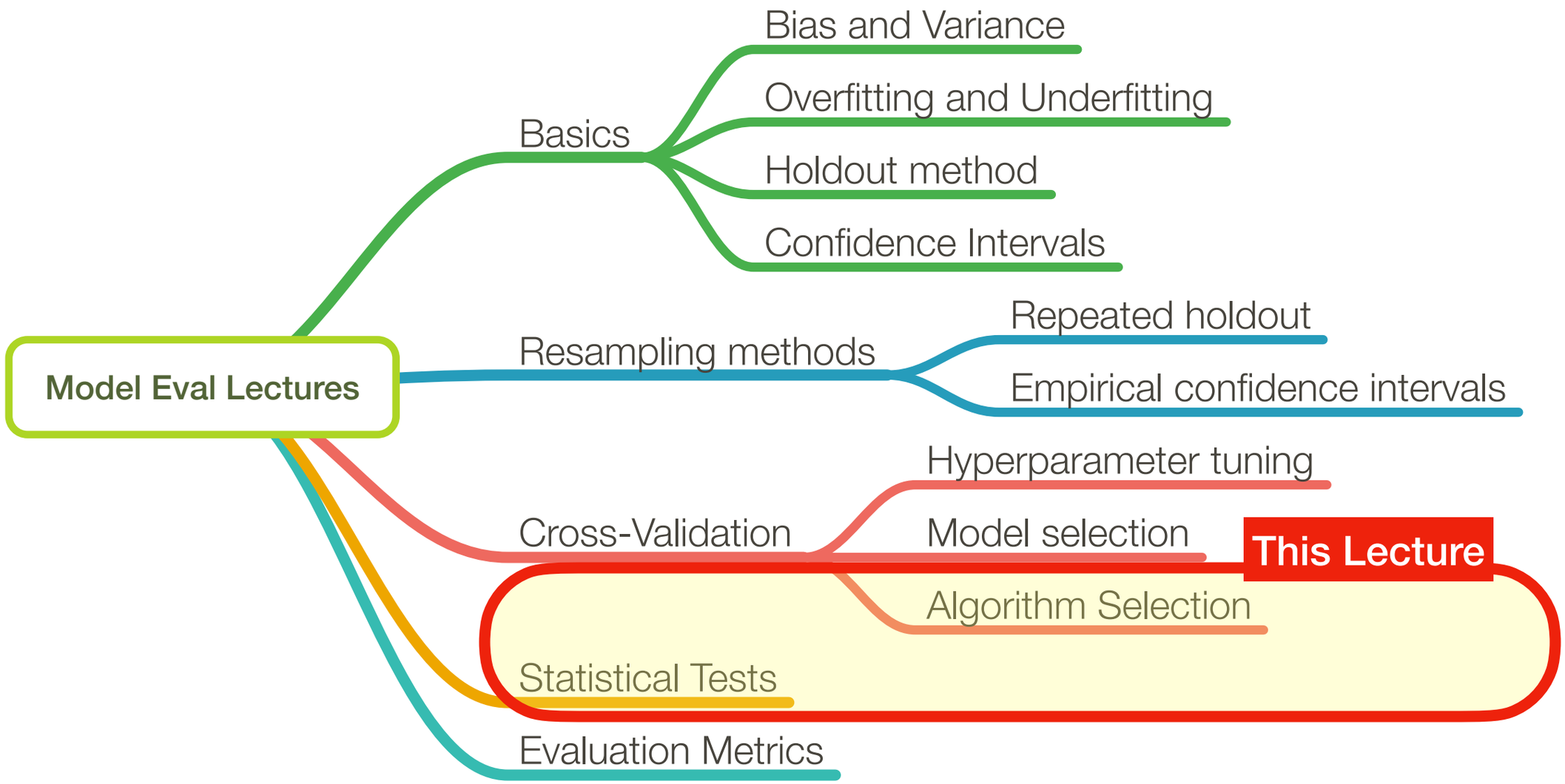
Model Evaluation 4: Algorithm Comparisons

STAT 451: Machine Learning, Fall 2021

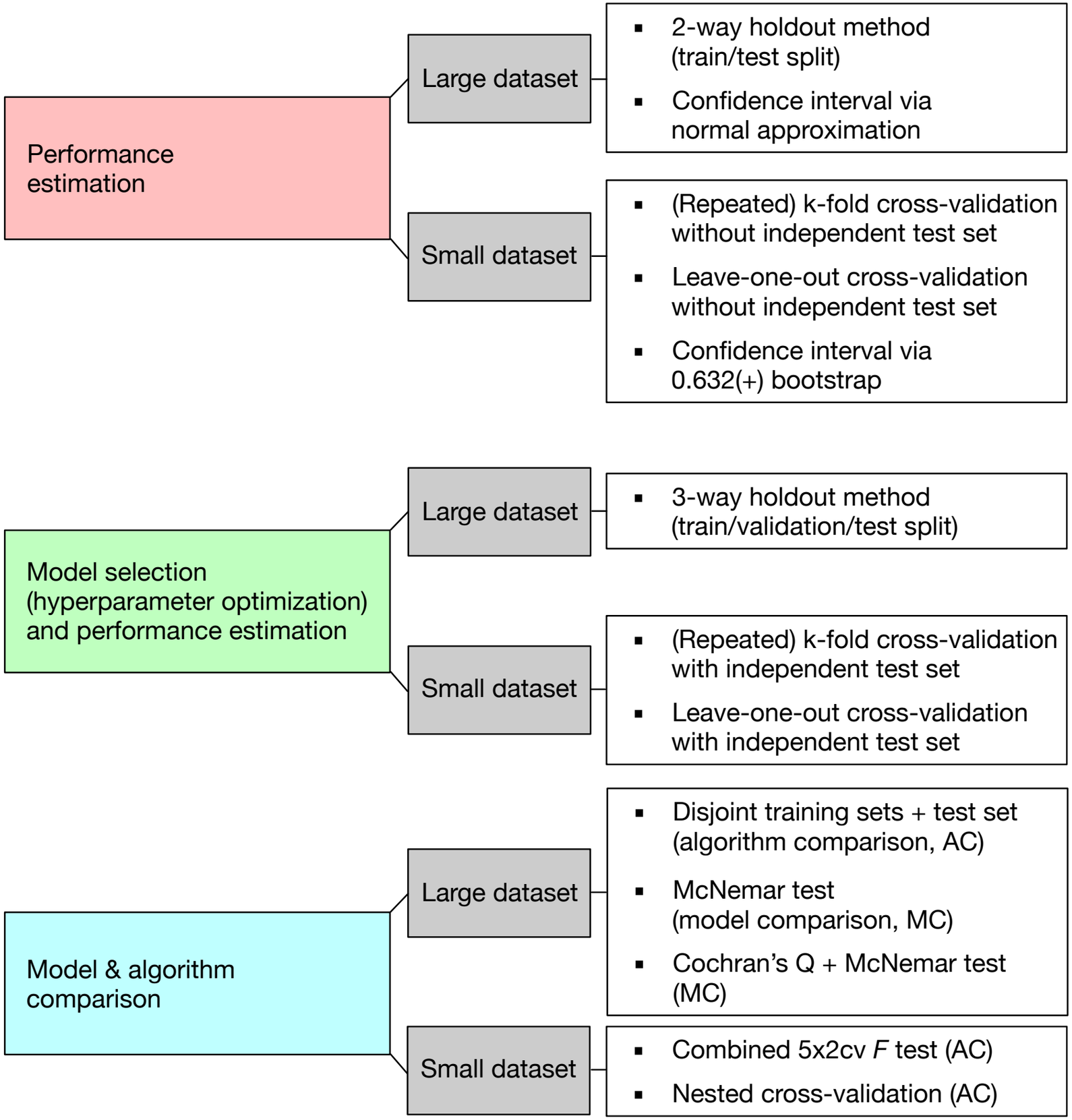
Sebastian Raschka

1. **Lecture Overview**
2. McNemar's Test
3. Multiple Pairwise Comparisons
4. Algorithm Selection (Statistical Inference)
5. Algorithm Selection (Computational/Empirical)
6. Nested CV Code Example

Overview


Model Eval Lectures
Basics
Bias and Variance
Overfitting and Underfitting
Holdout method
Confidence Intervals
Resampling methods
Repeated holdout
Empirical confidence intervals
Cross-Validation
Hyperparameter tuning
Model selection
Algorithm Selection
This Lecture
Statistical Tests
Evaluation Metrics

Overview, (my) "recommendations"



1. Lecture Overview

2. McNemar's Test

3. Multiple Pairwise Comparisons

4. Algorithm Selection (Statistical Inference)

5. Algorithm Selection (Computational/Empirical)

6. Nested CV Code Example

Comparing two machine learning classifiers -- McNemar's Test

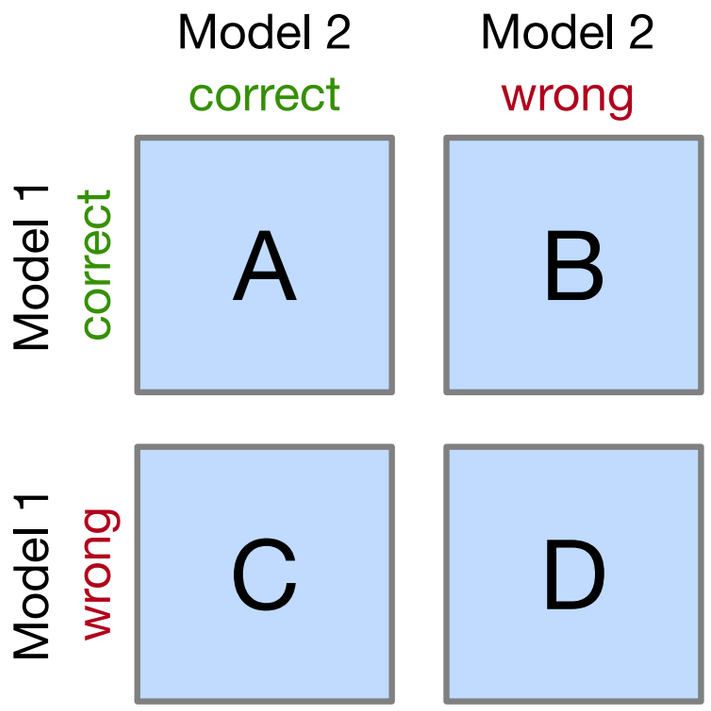
McNemar's test, introduced by Quinn McNemar in 1947 [1], is a non-parametric statistical test for paired comparisons that can be applied to compare the performance of two machine learning classifiers:

Task	Gaussian data	...	Paired nominal data
Compare a group to a reference value			Binomial test
Compare a pair of groups			McNemar's test
Compare two unpaired groups			χ^2 test, Fisher's exact test

[1] McNemar, Quinn. "Note on the sampling error of the difference between correlated proportions or percentages." *Psychometrika* 12.2 (1947): 153-157.

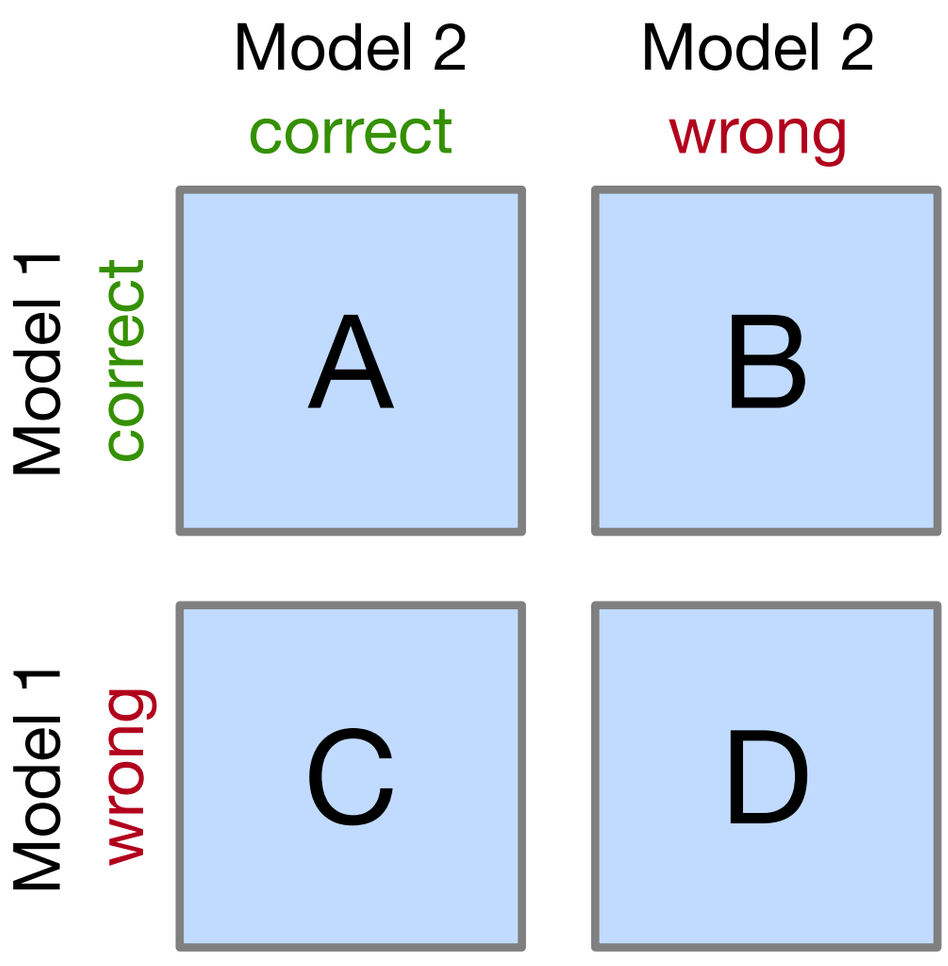
Comparing two machine learning classifiers -- McNemar's Test

- Also referred to as "within-subjects chi-squared test"
- Requires
  1) a categorical dependent variable with 2 categories (correct & incorrect)
  2) a categorical independent variable with two related groups (model 1 & model 2; paired through using same test set)
- Based on *a version of a* 2x2 confusion matrix
- Compares the predictions of two models to each other rather than listing false positive, true positive, false negative, and true negative counts of a single model
- The layout of the 2x2 confusion matrix suitable for McNemar's test is shown in the following figure:



Comparing two machine learning classifiers -- McNemar's Test

- Given such a 2x2 confusion matrix as shown in the previous figure, we can compute the accuracy of a *Model 1* via *(A+B) / (A+B+C+D)*
- Similarly, we can compute the accuracy of Model 2 as *(A+C) / N*
- Cells B and C (the off-diagonal entries) tell us how the models differ



Comparing two machine learning classifiers -- McNemar's Test

- Let's take a look at the following example:

A

		Model 2 correct	Model 2 wrong
Model 1	correct	9959	11
Model 1	wrong	1	29

B

		Model 2 correct	Model 2 wrong
Model 1	correct	9945	25
Model 1	wrong	15	15

- What is the prediction accuracy of models 1 and 2?

Comparing two machine learning classifiers -- McNemar's Test

- What is the prediction accuracy of models 1 and 2?

A

		Model 2 correct	Model 2 wrong
Model 1	correct	9959	**11**
Model 1	wrong	**1**	29

B

		Model 2 correct	Model 2 wrong
Model 1	correct	9945	**25**
Model 1	wrong	**15**	15

In both subpanel A and B, the accuracy of *Model 1* and *Model 2* are ???% and ???%, respectively.

- Model 1 accuracy subpanel A: $(???)/10000 \times 100\,\% = ???\,\%$
- Model 1 accuracy subpanel B: $(???)/10000 \times 100\,\% = ???\,\%$
- Model 2 accuracy subpanel A: $(???)/10000 \times 100\,\% = ???\,\%$
- Model 2 accuracy subpanel B: $(???)/10000 \times 100\,\% = ???\,\%$

Comparing two machine learning classifiers -- McNemar's Test

In both subpanel A and B, the accuracy of *Model 1* and *Model 2* are 99.7% and 99.6%, respectively.

A

		Model 2 correct	Model 2 wrong
Model 1	correct	9959	11
Model 1	wrong	1	29

B

		Model 2 correct	Model 2 wrong
Model 1	correct	9945	25
Model 1	wrong	15	15

In subpanel A:

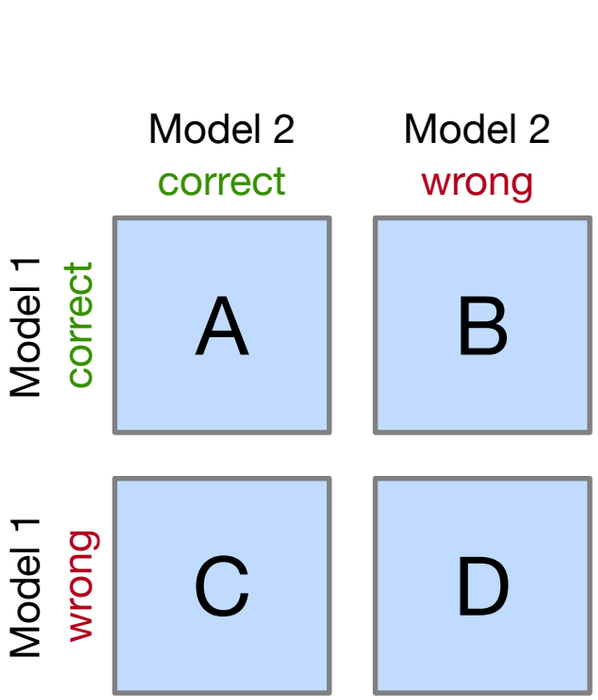
- *Model 1* got 11 predictions right that *Model 2* got wrong
- *Model 2* got 1 prediction right that *Model 1* got wrong
- Based on this 11:1 ratio (based on our intuition), does *Model 1* perform substantially better than *Model 2?*

In subpanel B:

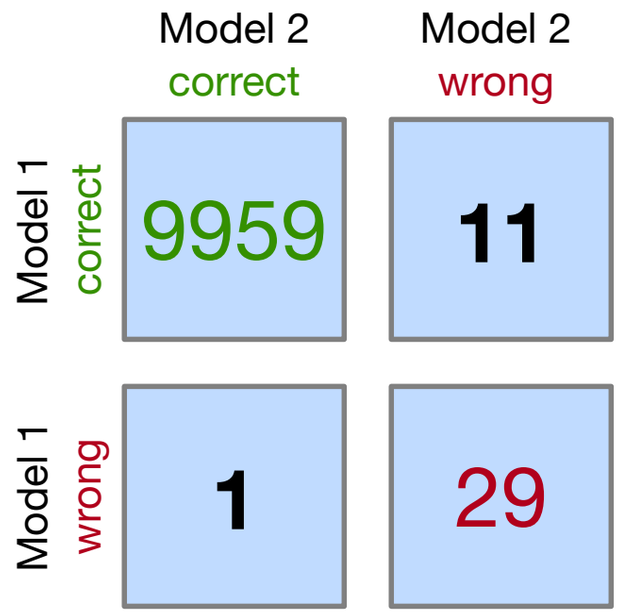
- The *Model 1*:*Model 2* ratio is 25:15
- This is less conclusive about which model is the better one to choose.

Comparing two machine learning classifiers -- McNemar's Test

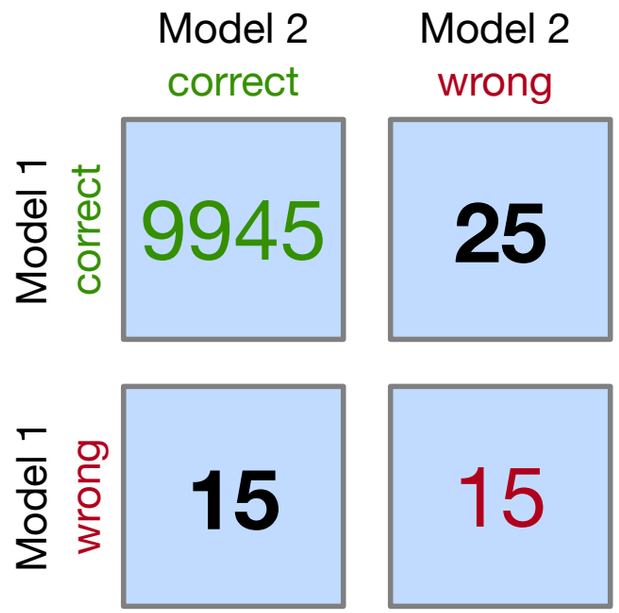
In both subpanel A and B, the accuracy of *Model 1* and *Model 2* are 99.7% and 99.6%, respectively.



A



B

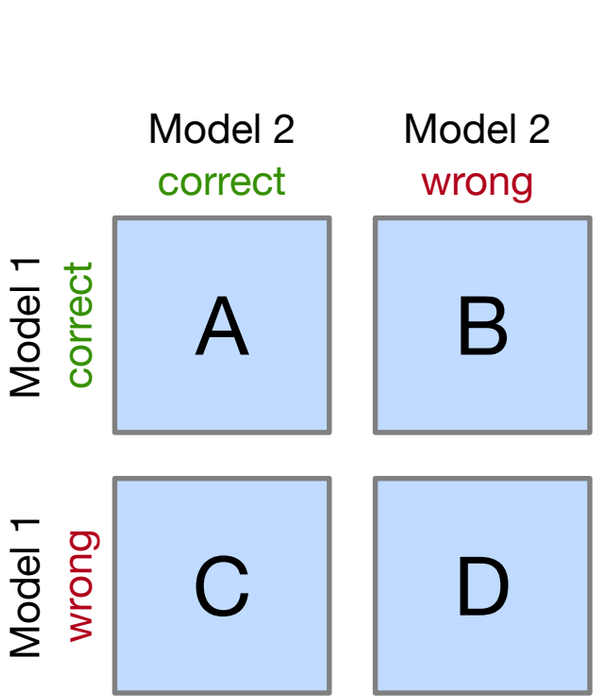


In McNemar's Test, we formulate the

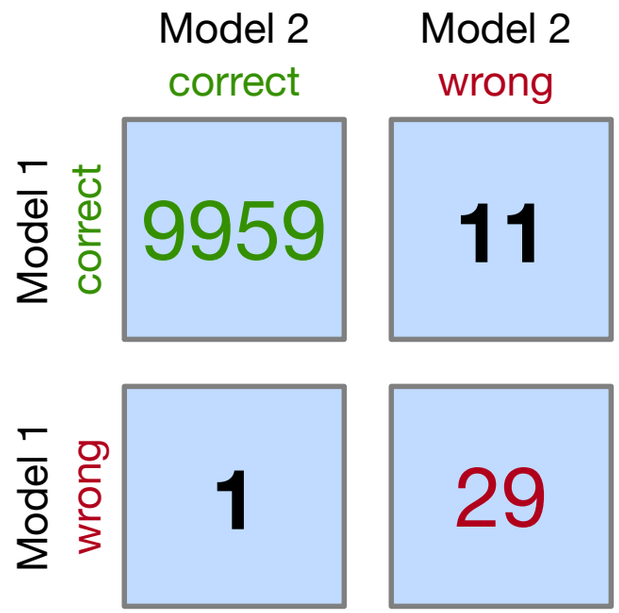
- null hypothesis: the probabilities *p(B*) and *p(C)* are the same
- alternative hypothesis: the performances of the two models are not equal

Comparing two machine learning classifiers -- McNemar's Test

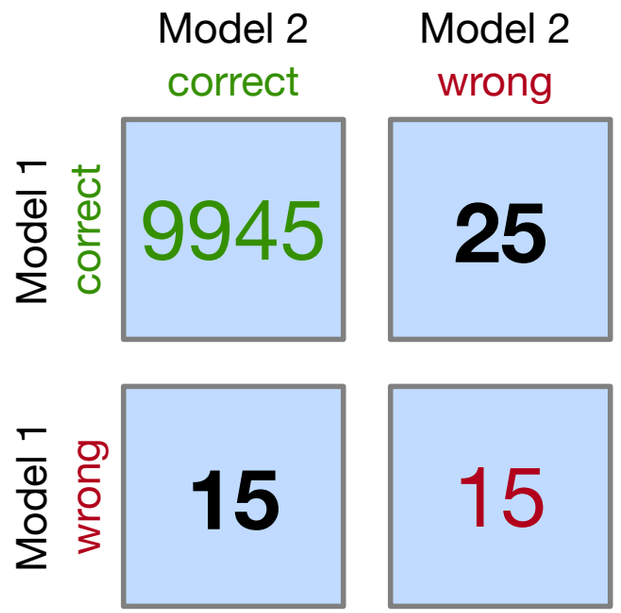
In both subpanel A and B, the accuracy of *Model 1* and *Model 2* are 99.7% and 99.6%, respectively.



A



B



In McNemar's Test, we formulate the

- null hypothesis: the probabilities *p(B*) and *p(C)* are the same
- alternative hypothesis: the performances of the two models are not equal

The McNemar test statistic ("chi-squared") can be computed as follows:

$$\chi^2 = \frac{(B - C)^2}{B + C}$$

Comparing two machine learning classifiers -- McNemar's Test

The McNemar test statistic ("chi-squared") can be computed as follows:

$$\chi^2 = \frac{(B - C)^2}{B + C}$$

- Set a significance threshold, for example, $\alpha = 0.05$
- Compute the p-value -- assuming that the null hypothesis is true, the p-value is the probability of observing the given empirical (or a larger) chi-squared value (chi^2 distribution with 1 degree of freedom, and relatively large numbers in cells B and C, say > 25)
- If the p-value is lower than our chosen significance level, we can reject the null hypothesis that the two model's performances are equal

Comparing two machine learning classifiers -- McNemar's Test

A

		Model 2 correct	Model 2 wrong
Model 1	correct	9959	11
Model 1	wrong	1	29

B

		Model 2 correct	Model 2 wrong
Model 1	correct	9945	25
Model 1	wrong	15	15

- If we did this for scenario B in the previous figure (chi^2=2.5), we would obtain a p-value of 0.1138, which is larger than our significance threshold, and thus, we cannot reject the null hypothesis.
- If we computed the p-value for scenario A (chi^2=8.3), we would obtain a p-value of 0.0039, which is below the set significance threshold (alpha=0.05) and leads to the rejection of the null hypothesis; we can conclude that the models' performances are different (for instance, Model 1 performs better than Model 2).

Comparing two machine learning classifiers -- McNemar's Test

Continuity Correction

Approximately 1 year after Quinn McNemar published the McNemar Test (McNemar 1947), Allen L. Edwards [1] proposed a continuity corrected version, which is the more commonly used variant today:

$$\chi^2 = \frac{\left(|B - C| - 1 \right)^2}{B + C}.$$

"This correction will have the obvious result of reducing the absolute value of the difference, [B - C], by unity." [1]

[1] Edwards, Allen L. "Note on the “correction for continuity” in testing the significance of the difference between correlated proportions." *Psychometrika* 13.3 (1948): 185-187.

Comparing two machine learning classifiers -- McNemar's Test

Exact p-values via the Binomial test

- McNemar's test approximates the p-values reasonably well if the values in cells B and C are larger than 50
- But it makes sense to use a computationally more expensive binomial test to compute the exact p-values (esp. if B and C are relatively small) -- since the chi-squared value from McNemar's test may not be well-approximated by the chi-squared distribution

Comparing two machine learning classifiers -- McNemar's Test

Exact p-values via the Binomial test

- McNemar's test approximates the p-values reasonably well if the values in cells B and C are larger than 50
- But it makes sense to use a computationally more expensive binomial test to compute the exact p-values (esp. if B and C are relatively small) -- since the chi-squared value from McNemar's test may not be well-approximated by the chi-squared distribution

The exact p-value can be computed as follows:

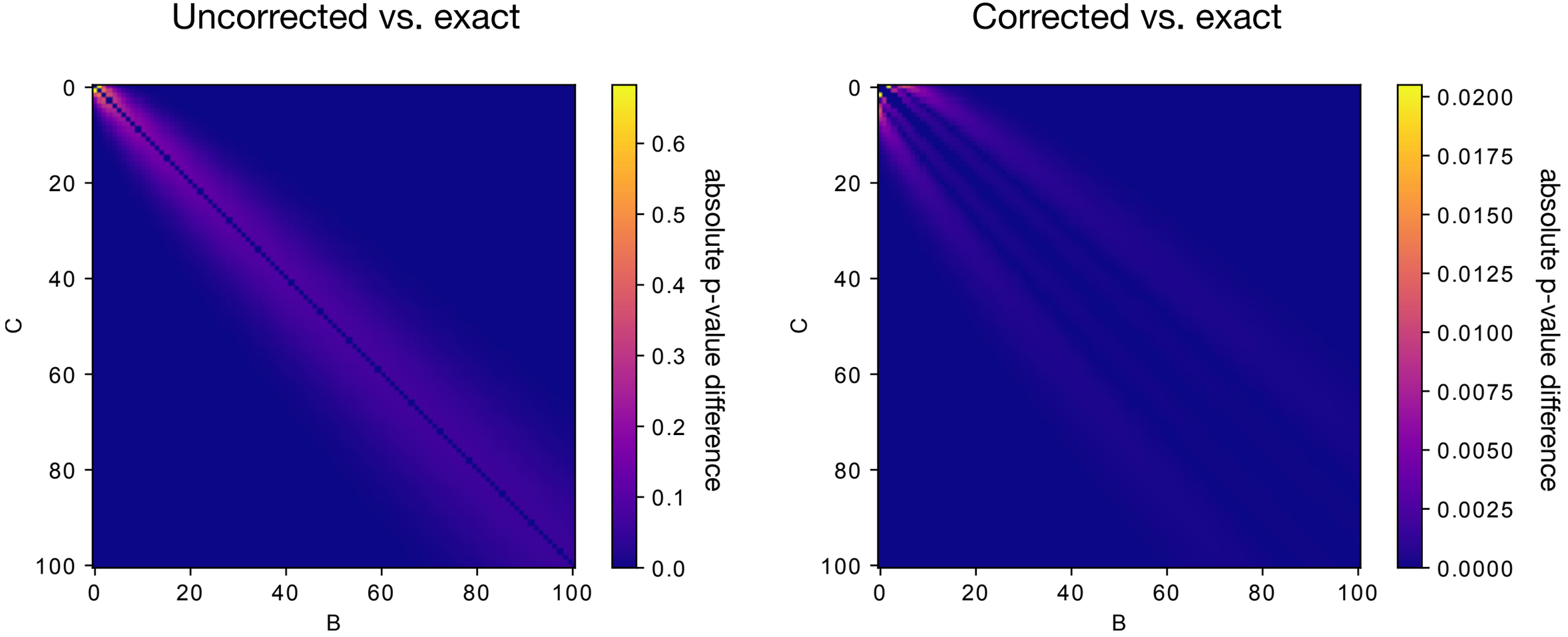
$$p = 2 \sum_{i=max(B,C)}^{n} \binom{n}{i} 0.5^i (1 - 0.5)^{n-i},$$

where n=b+c, and the factor 2 is used to compute the two-sided p-value.

Comparing two machine learning classifiers -- McNemar's Test

Exact p-values via the Binomial test

- The following heat map illustrates the differences between the McNemar approximation of the chi-squared value (with and without Edward's continuity correction) to the exact p-values computed via the binomial test:



(As we can see in this heat map, the p-values from the continuity-corrected version of McNemar's test are almost identical to the p-values from a binomial test if both B and C are larger than 50.)

McNemar test implementation:
http://rasbt.github.io/mlxtend/user_guide/evaluate/mcnemar/

API

mcnemar(ary, corrected=True, exact=False)

McNemar test for paired nominal data

Parameters

- `ary` : array-like, shape=[2, 2]

  2 x 2 contigency table (as returned by evaluate.mcnemar_table), where a: ary[0, 0]: # of samples that both models predicted correctly b: ary[0, 1]: # of samples that model 1 got right and model 2 got wrong c: ary[1, 0]: # of samples that model 2 got right and model 1 got wrong d: aryCell [1, 1]: # of samples that both models predicted incorrectly

- `corrected` : array-like, shape=[n_samples] (default: True)

  Uses Edward's continuity correction for chi-squared if `True`

- `exact` : bool, (default: False)

  If `True`, uses an exact binomial test comparing b to a binomial distribution with n = b + c and p = 0.5. It is highly recommended to use `exact=True` for sample sizes < 25 since chi-squared is not well-approximated by the chi-squared distribution!

Returns

- `chi2, p` : float or None, float

  Returns the chi-squared value and the p-value; if `exact=True` (default: `False`), `chi2` is `None`

1. Lecture Overview
2. McNemar's Test
3. **Multiple Pairwise Comparisons**
4. Algorithm Selection (Statistical Inference)
5. Algorithm Selection (Computational/Empirical)
6. Nested CV Code Example

Multiple Hypothesis Testing Issue

1. Conduct an omnibus test under the null hypothesis that there is no difference between the classification accuracies

2. If the omnibus test led to the rejection of the null hypothesis, conduct pairwise post hoc tests, with adjustments for multiple comparisons, to determine where the differences between the model performances occurred

Multiple Hypothesis Testing Issue

1. Conduct an omnibus test under the null hypothesis that there is no difference between the classification accuracies (Cochran's Q test would be a good choice, which is a generalized version of McNemar's test for three or more models)

2. If the omnibus test led to the rejection of the null hypothesis, conduct pairwise post hoc tests, with adjustments for multiple comparisons, to determine where the differences between the model performances occurred (McNemar's Test would be a candidate here)

Cochran's Q Test

- Cochran's Q test is analogous to ANOVA for binary outcomes

- The test statistic is approximately (similar to McNemar's test) distributed as chi-squared with *M*–1 degrees of freedom, where L is the number of models we evaluate (since *M*=2 for McNemar's test, McNemar's test statistic approximates a chi-squared distribution with one degree of freedom)

More formally, Cochran's Q test tests the hypothesis that there is no difference between the classification accuracies

$$H_0 : ACC_1 = ACC_2 = \ldots = ACC_M$$

http://rasbt.github.io/mlxtend/user_guide/evaluate/cochrans_q/

mlxtend Home User Guide API Installation About Search GitHub

Cochran's Q Test

Cochran's Q test for comparing the performance of multiple classifiers.

```
from mlxtend.evaluate import cochrans_q
```

Overview

Cochran's Q test can be regarded as a generalized version of McNemar's test that can be applied to evaluate multiple classifiers. In a sense, Cochran's Q test is analogous to ANOVA for binary outcomes.

To compare more than two classifiers, we can use Cochran's Q test, which has a test statistic Q that is approximately, (similar to McNemar's test), distributed as chi-squared with $L - 1$ degrees of freedom, where L is the number of models we evaluate (since $L = 2$ for McNemar's test, McNemars test statistic approximates a chi-squared distribution with one degree of freedom).

More formally, Cochran's Q test tests the hypothesis that there is no difference between the classification accuracies [1]:

$$p_i : H_0 = p_1 = p_2 = \cdots = p_L.$$

Let $\{D_1, \ldots, D_L\}$ be a set of classifiers who have all been tested on the same dataset. If the L classifiers don't perform differently, then the following Q statistic is distributed approximately as "chi-squared" with $L - 1$ degrees of freedom:

$$Q_C = (L-1)\frac{L\sum_{i=1}^{L} G_i^2 - T^2}{LT - \sum_{j=1}^{N_{ts}}(L_j)^2}.$$

Here, G_i is the number of objects out of N_{ts} correctly classified by $D_i = 1, \ldots L$; L_j is the number of classifiers out of L that correctly classified object $\mathbf{z}_j \in \mathbf{Z}_{ts}$, where $\mathbf{Z}_{ts} = \{\mathbf{z}_1, \ldots \mathbf{z}_{N_{ts}}\}$ is the test dataset on which the classifers are tested on; and T is the total number of correct number of votes among the L classifiers [2]:

$$T = \sum_{i=1}^{L} G_i = \sum_{j=1}^{N_{ts}} L_j.$$

1. Lecture Overview
2. McNemar's Test
3. Multiple Pairwise Comparisons
4. **Algorithm Selection (Statistical Inference)**
5. Algorithm Selection (Computational/Empirical)
6. Nested CV Code Example

Algorithm Selection

Aside from publishing papers,
what would be a real-world application
(vs. model evaluation)?

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, *10*(7), 1895-1923:

Summary:

1. McNemar's test
   - low false positive rate
   - fast, only needs to be executed once

2. Difference in proportions, by Snedecor and Cochran
   - high false positive rate (here, incorrectly detect difference when there is none)
   - cheap to compute though

3. Resampled paired t-test
   - high false positive rate
   - computationally very expensive

4. k-fold cross-validated t-test
   - somewhat elated false positive rate

5. 5x2cv paired t-test
   - low false positive rate (similar to McNemarr)
   - slightly more powerful than McNemar; recommended if computational efficiency (runtime) is not an issue (10 times more computations than McNemar)

Optional:

For more information on statistical tests, see lecture notes

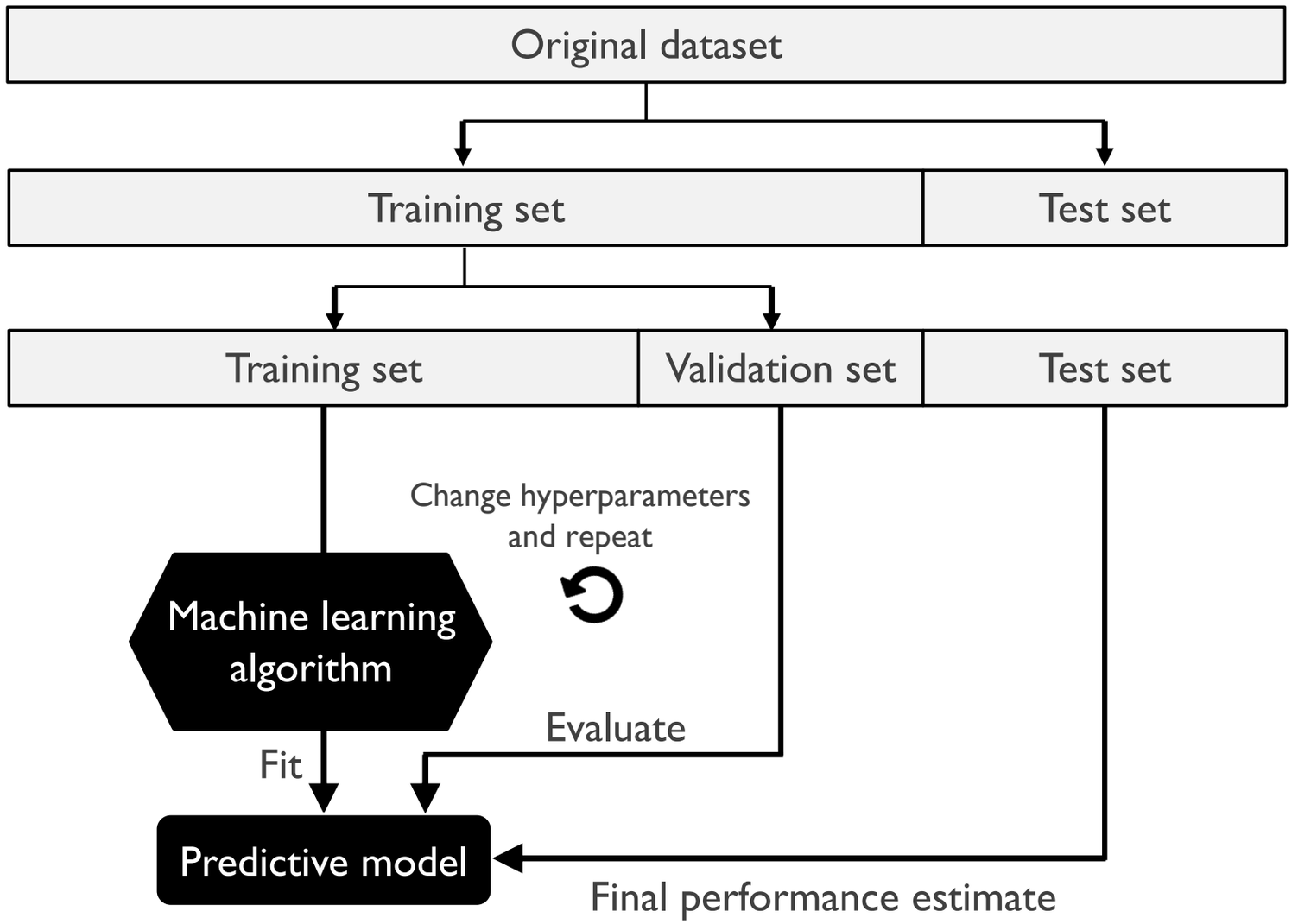
Code Examples

- McNemar's Test http://rasbt.github.io/mlxtend/user_guide/evaluate/mcnemar/
- Cochran's Q Test http://rasbt.github.io/mlxtend/user_guide/evaluate/cochrans_q/
- Resampled paired *t* test http://rasbt.github.io/mlxtend/user_guide/evaluate/paired_ttest_resampled/
- K-fold cross-validated paired *t* test http://rasbt.github.io/mlxtend/user_guide/evaluate/paired_ttest_kfold_cv/
- 5x2cv paired *t* test http://rasbt.github.io/mlxtend/user_guide/evaluate/paired_ttest_5x2cv/
- 5x2cv combined *F* test http://rasbt.github.io/mlxtend/user_guide/evaluate/combined_ftest_5x2cv/

1. Lecture Overview
2. McNemar's Test
3. Multiple Pairwise Comparisons
4. Algorithm Selection (Statistical Inference)
5. **Algorithm Selection (Computational/Empirical)**
6. Nested CV Code Example

Back to "Computational/Empirical" Methods

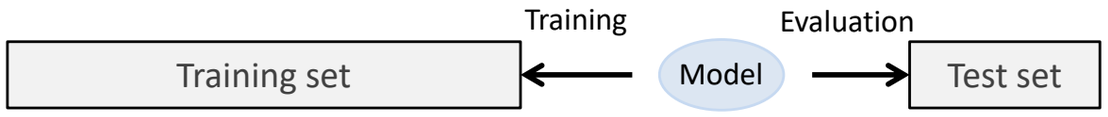
Recap: Model Selection with 3-way Holdout



Recap: Model Selection with k-fold Cross.-Val.

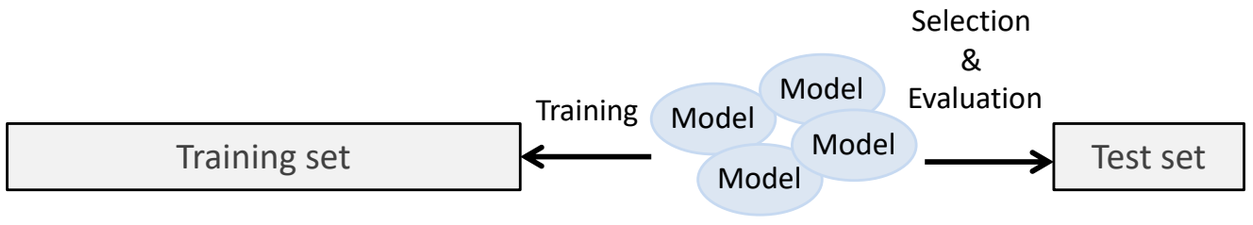
1)

good or bad ?



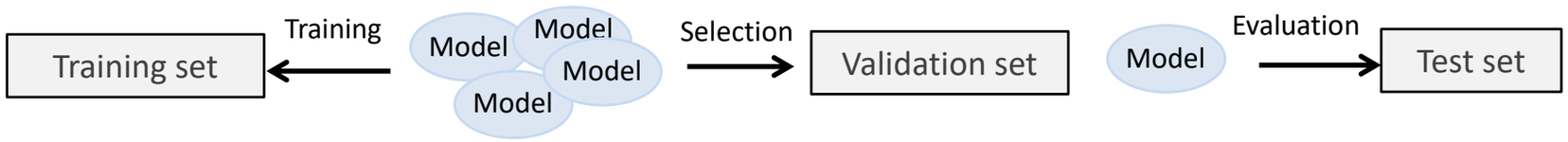
2)

good or bad ?



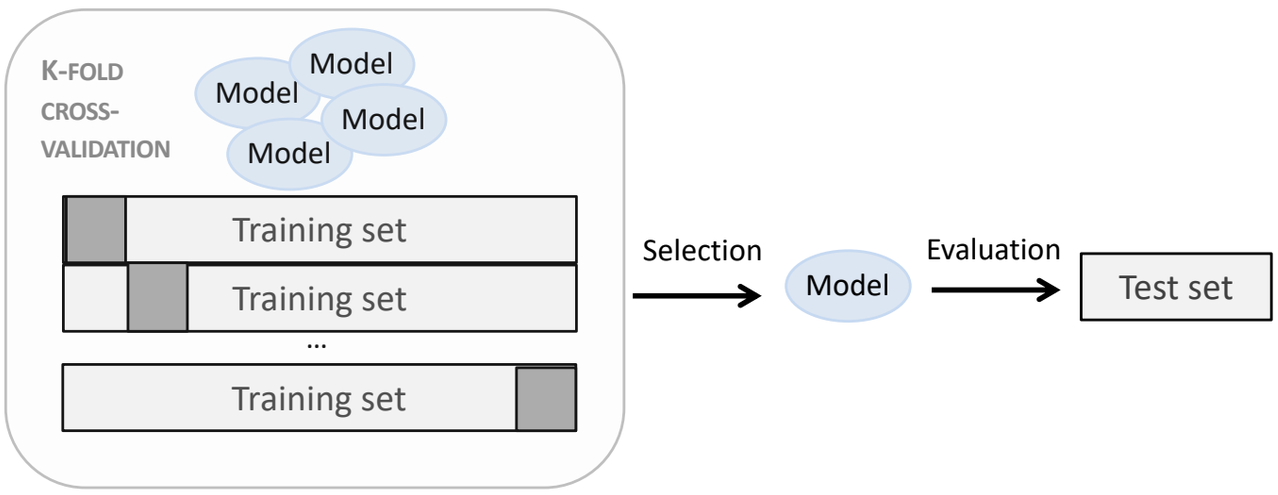
3)

good or bad ?



4)

good or bad ?

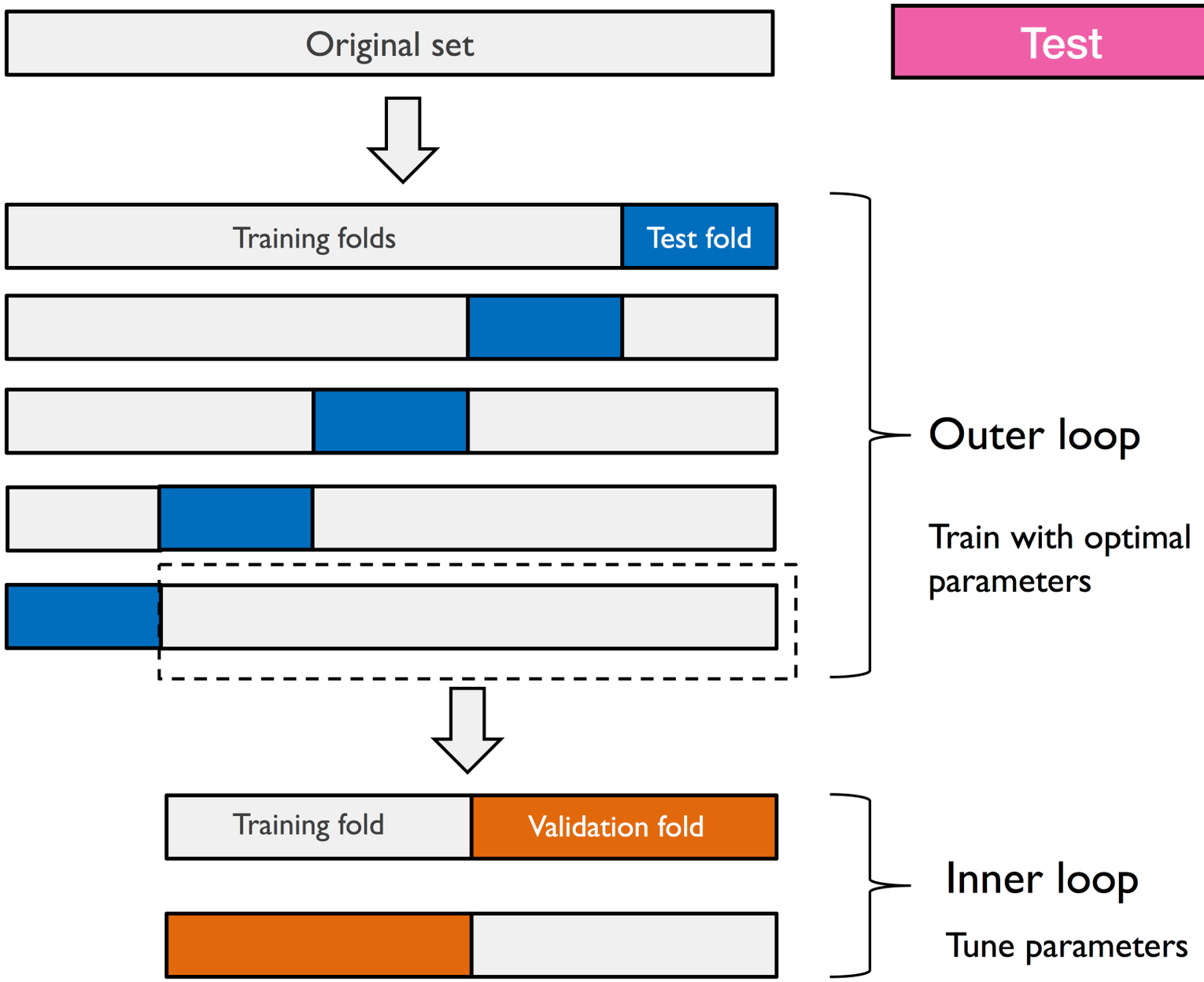


Nested Cross-Validation for Algorithm Selection

Main Idea:

- Outer loop: purpose related to train/test split
- Inner loop: like k-fold cross-validation for tuning

Nested Cross-Validation



Nested Cross-Validation for Algorithm Selection

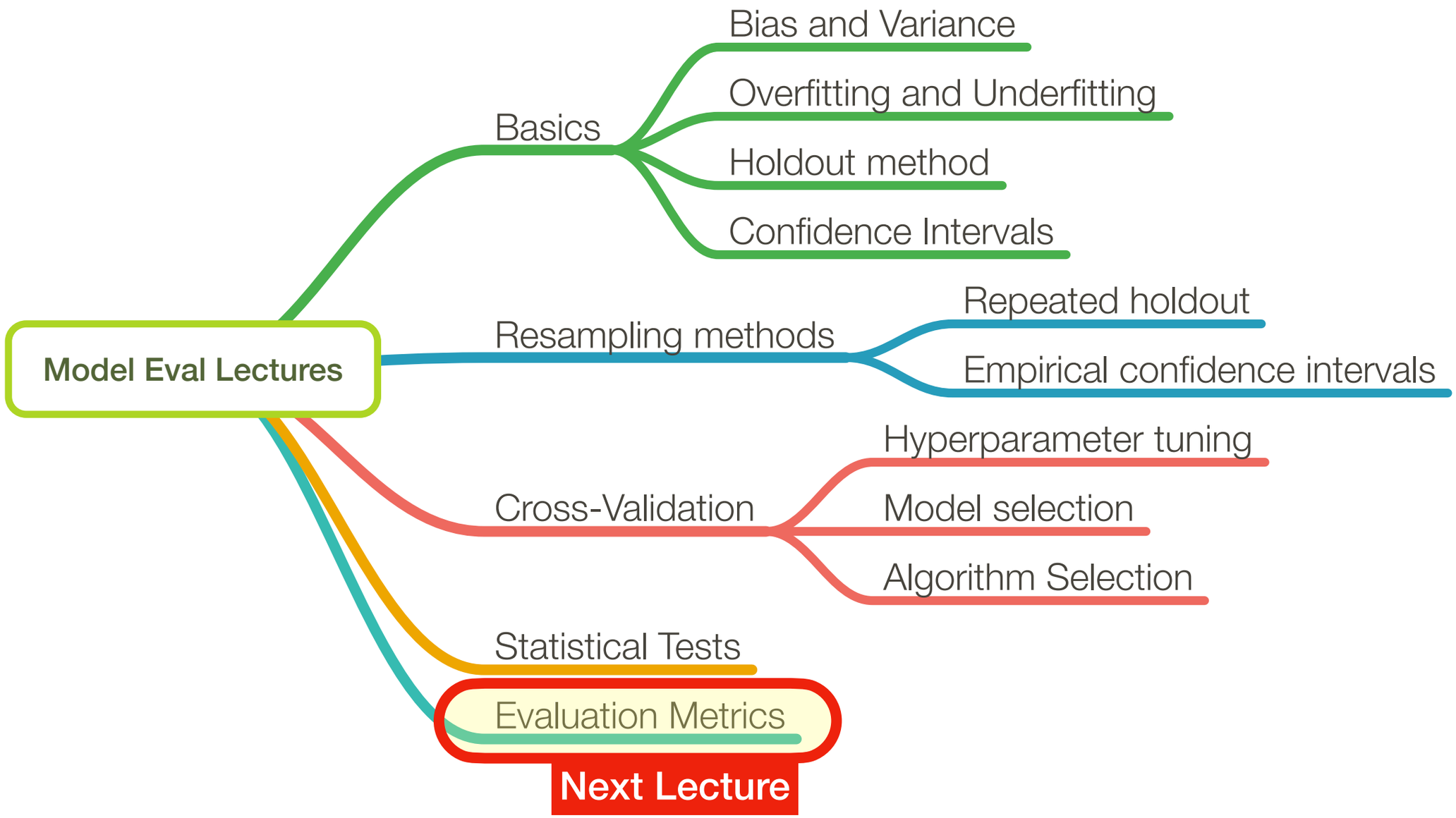
- Outer loop:
  use average performance as generalization performance
  check for "model stability"

- Finally:
  as usual, fit model on whole dataset for deployment

1. Lecture Overview
2. McNemar's Test
3. Multiple Pairwise Comparisons
4. Algorithm Selection (Statistical Inference)
5. Algorithm Selection (Computational/Empirical)
6. **Nested CV Code Example**

https://github.com/rasbt/stat451-machine-learning-fs20/tree/master/L11/code

Overview


Model Eval Lectures
Basics
Bias and Variance
Overfitting and Underfitting
Holdout method
Confidence Intervals
Resampling methods
Repeated holdout
Empirical confidence intervals
Cross-Validation
Hyperparameter tuning
Model selection
Algorithm Selection
Statistical Tests
Evaluation Metrics
Next Lecture