Lecture 11

Model Evaluation 4: Algorithm Comparisons

STAT 479: Machine Learning, Fall 2019

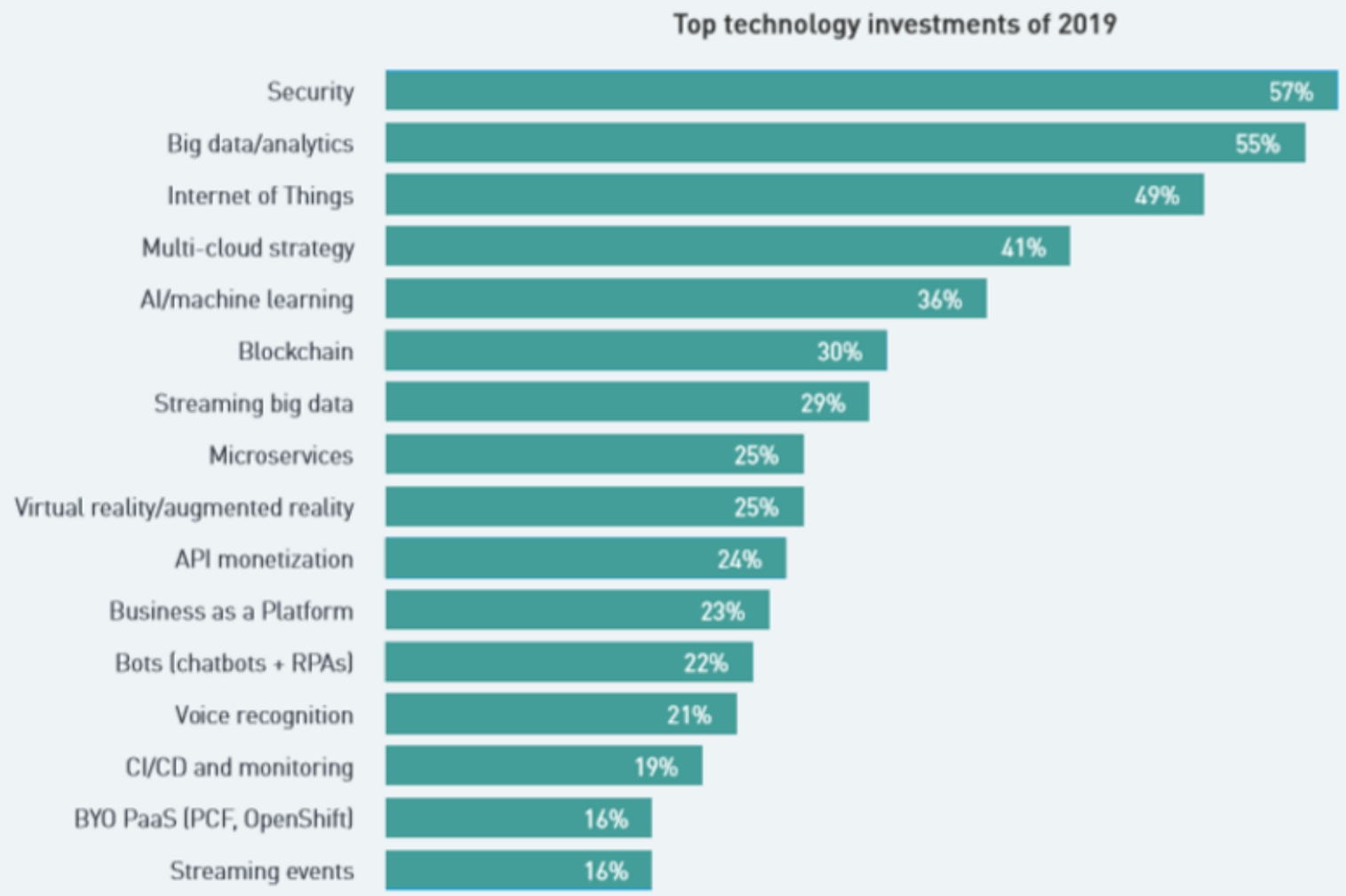
Sebastian Raschka

http://stat.wisc.edu/~sraschka/teaching/stat479-fs2019/

Some News and interesting tidbits

TREND 05 | PARTNERING WITH IT TO TURBOCHARGE THE BUSINESS

As organizations invest more in distributed and emerging technologies



Source: MuleSoft

https://www.zdnet.com/article/top-7-digital-transformation-trends-shaping-2020/

Artificial Intelligence / Machine Learning

DeepMind's AI has now outcompeted nearly all human players at StarCraft II

AlphaStar cooperated with itself to learn new strategies for conquering the popular galactic warfare game.

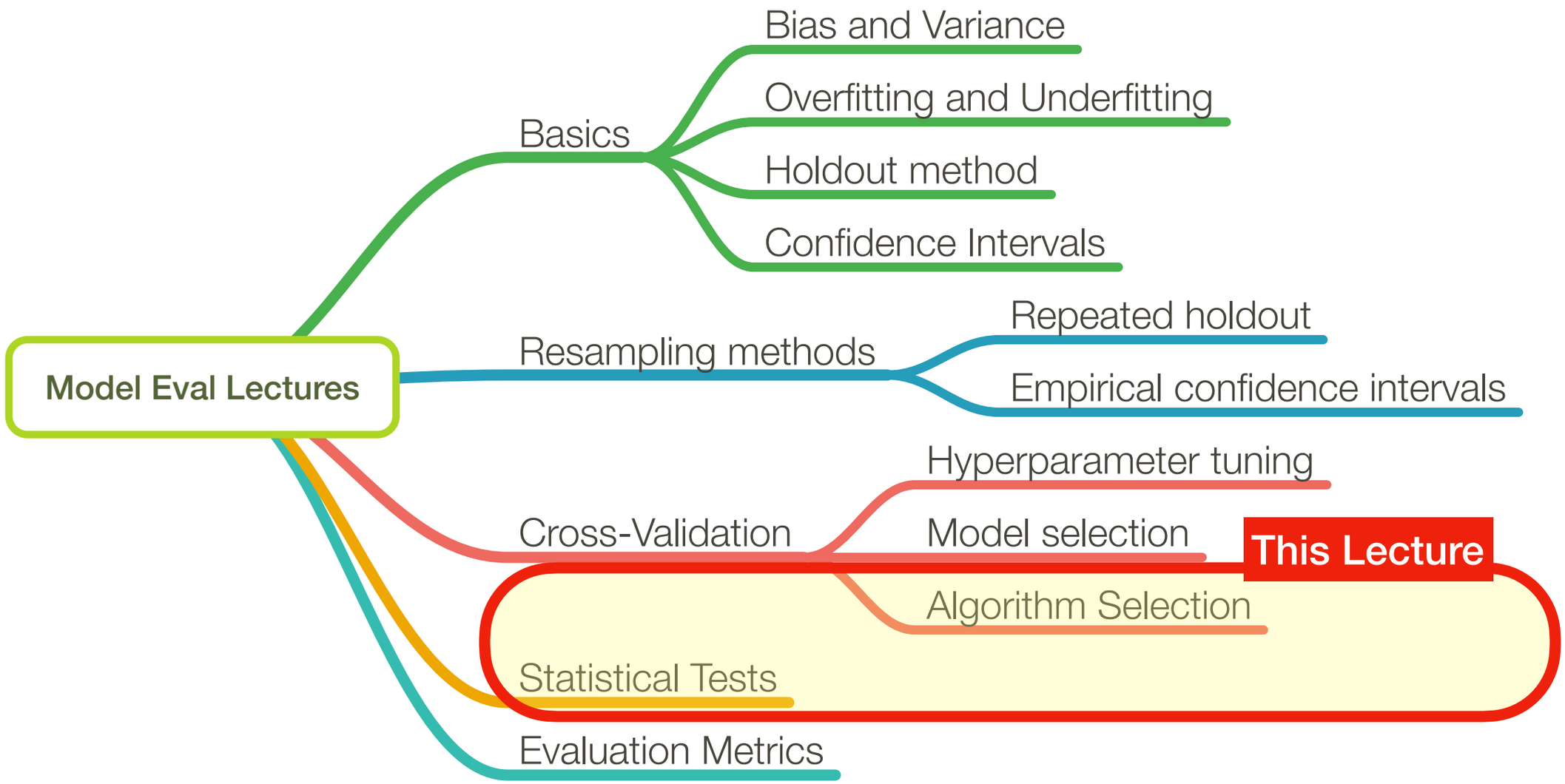
by **Karen Hao** Oct 30, 2019

In order to attain such flexibility, the DeepMind team modified a commonly used technique known as self-play, in which a reinforcement-learning algorithm plays against itself to learn faster. DeepMind famously used this technique to train AlphaGo Zero, the program that taught itself without any human input to beat the best players in the ancient game of Go. The lab also used it in the preliminary version of AlphaStar.

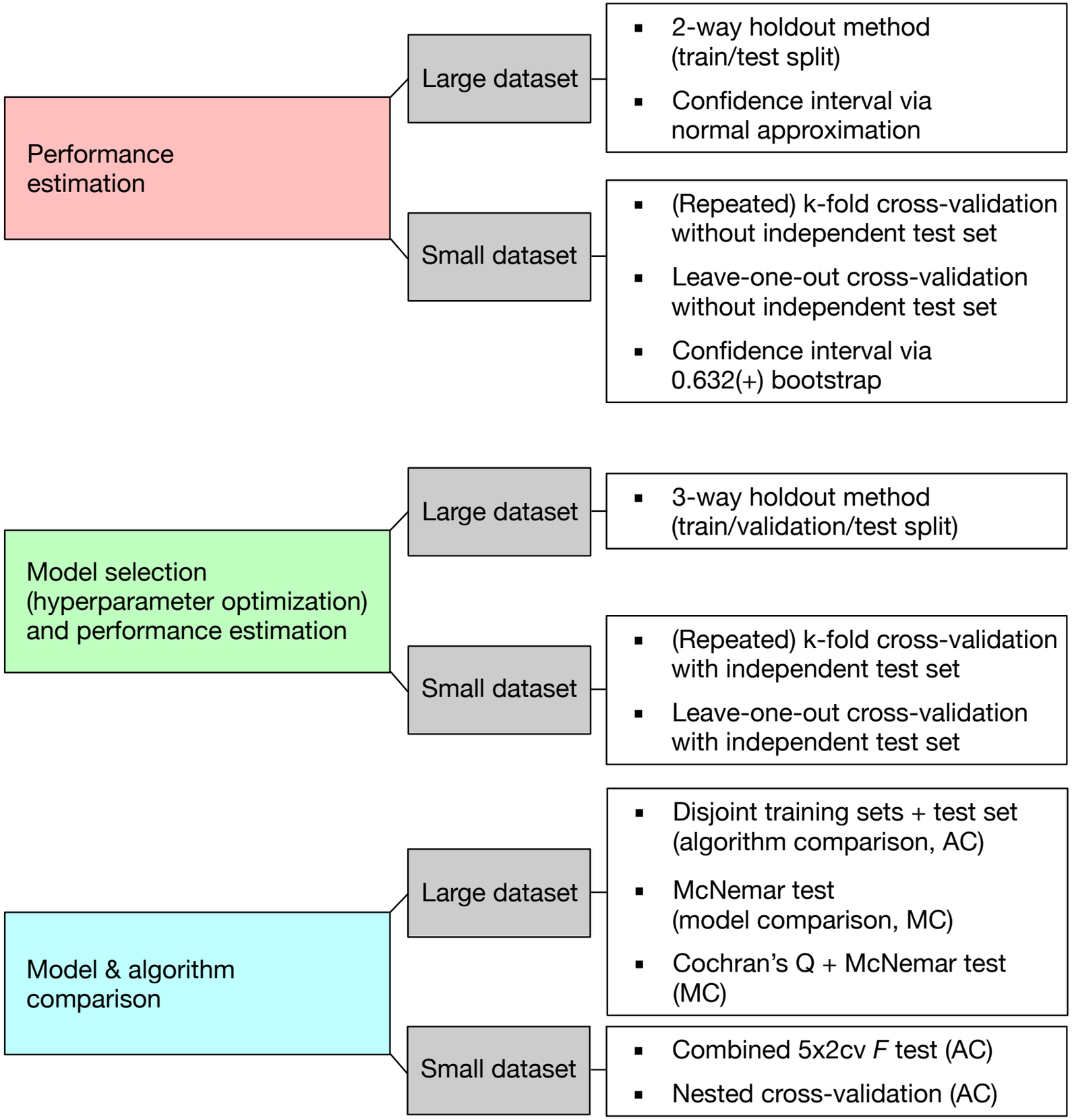
https://www.technologyreview.com/s/614650/ai-deepmind-outcompeted-most-players-at-starcraft-ii/

https://stroemer.cc/resample-imbalanced-data/

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"



Comparing two machine learning classifiers -- McNemar's Test

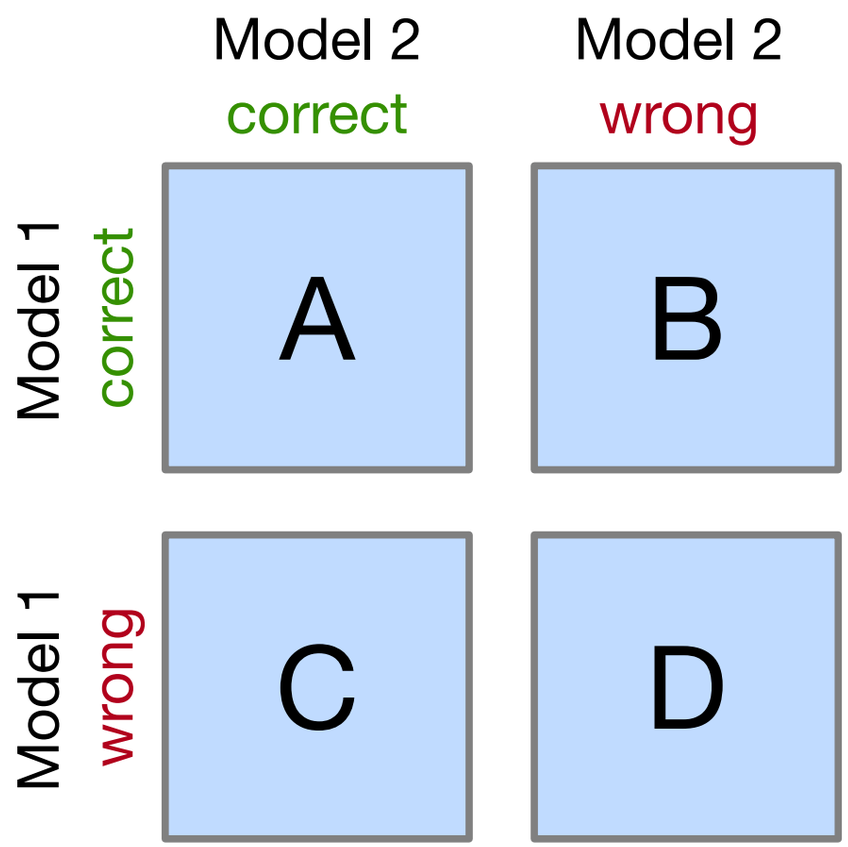
McNemar's test, introduced by Quinn McNemar in 1947 [1], is a non-parametric statistical test for paired comparisons that can be applied to compare the performance of two machine learning classifiers:

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.

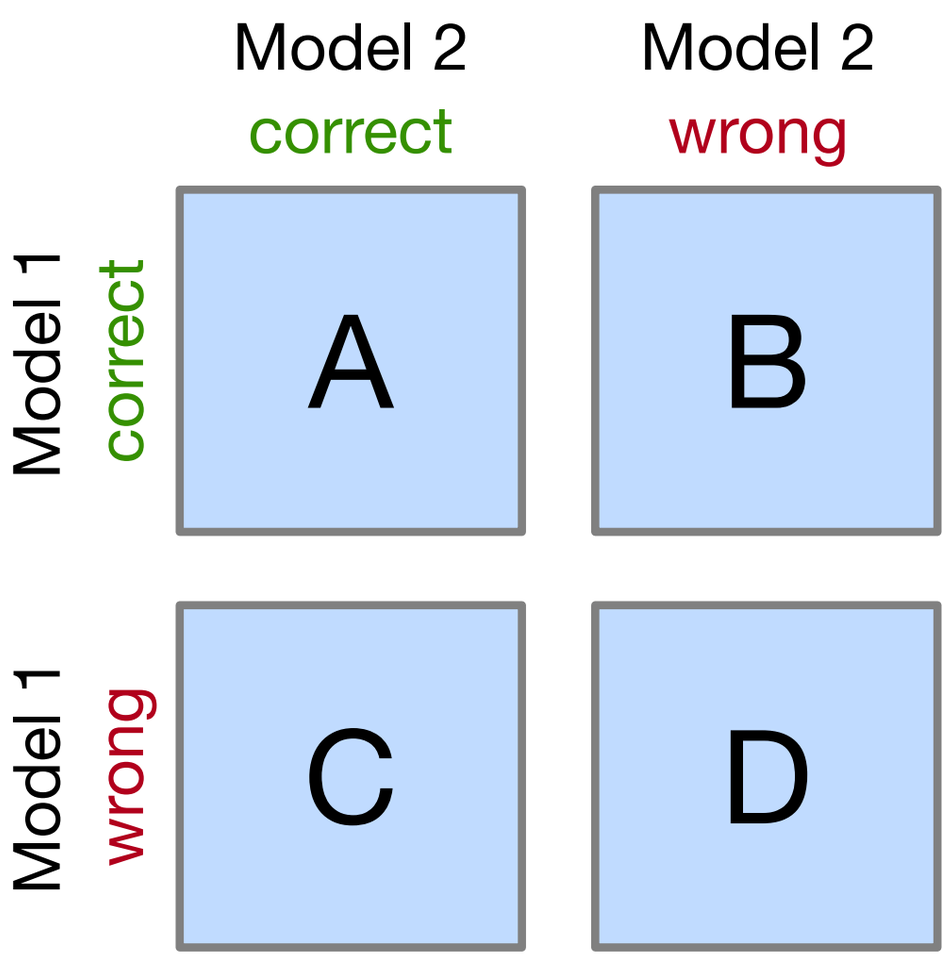
Comparing two machine learning classifiers -- McNemar's Test

- Also referred to as "within-subjects chi-squared test"
- Applied to paired nominal data 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:

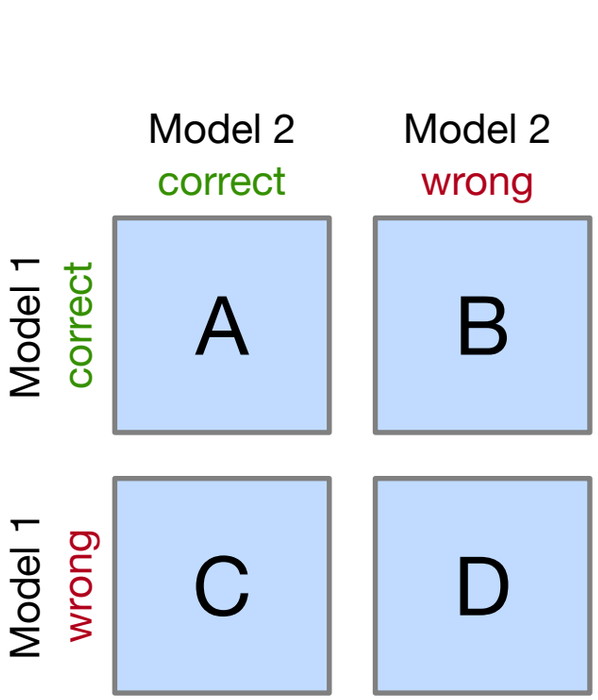
- *Model 1* got 11 predictions right that *Model 2* got wrong
- *Model 2* got 1 prediction right that *Model 1* got wrong
- Based on this 11:1 ratio (based on our intuition), does *Model 1* perform substantially better than *Model 2?*

In subpanel B:

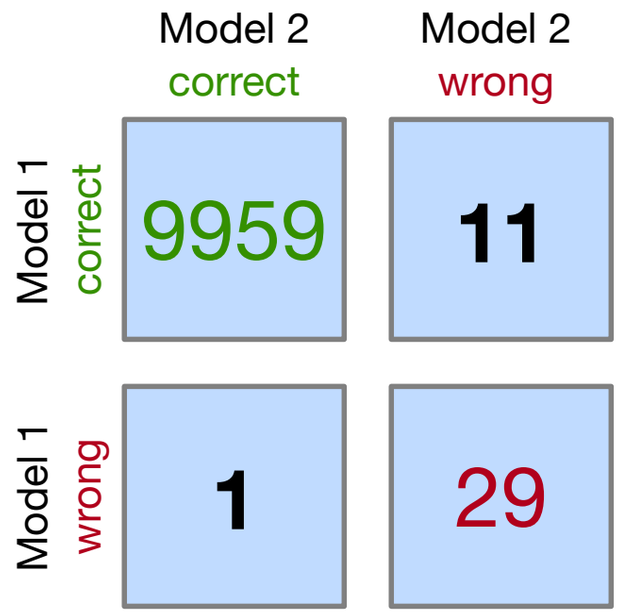
- The *Model 1*:*Model 2* ratio is 25:15
- This is less conclusive about which model is the better one to choose.

Comparing two machine learning classifiers -- McNemar's Test

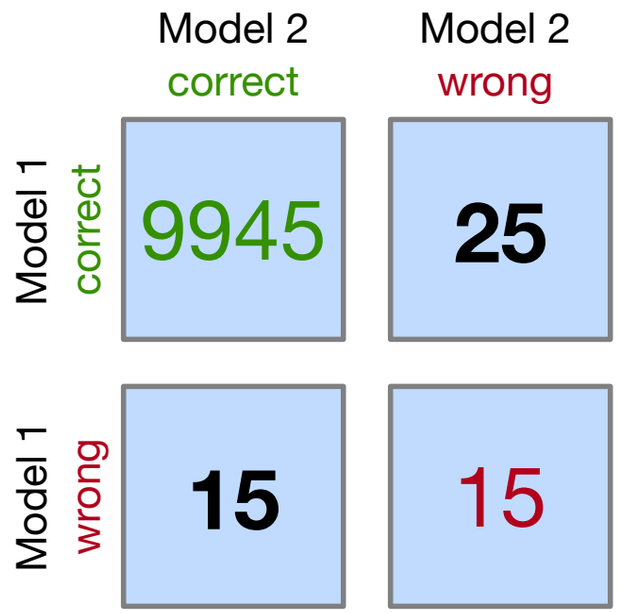
In both subpanel A and B, the accuracy of *Model 1* and *Model 2* are 99.7% and 99.6%, respectively.



A



B

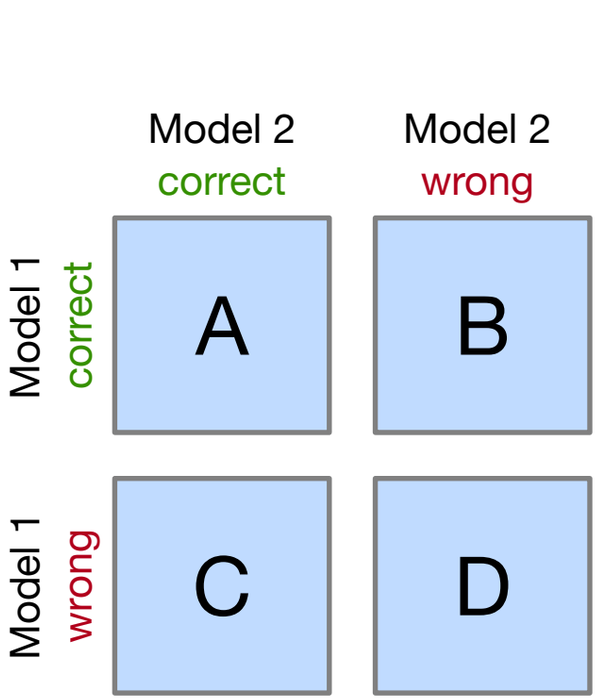


In McNemar's Test, we formulate the

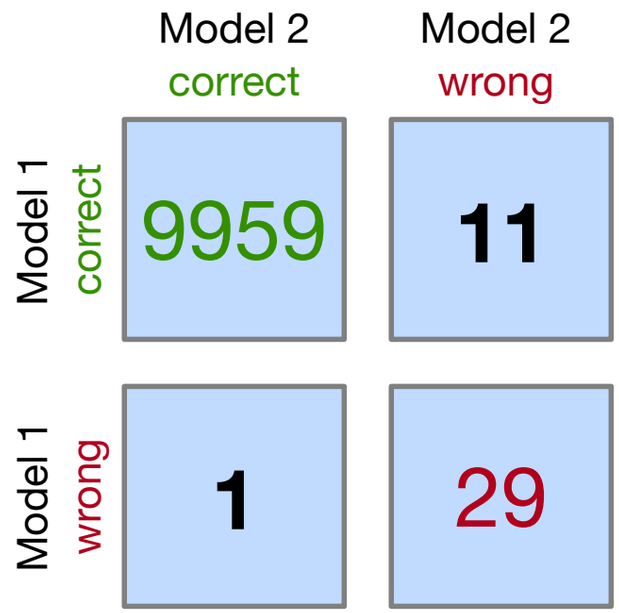
- null hypothesis: the probabilities *p(B*) and *p(C)* are the same
- alternative hypothesis: the performances of the two models are not equal

Comparing two machine learning classifiers -- McNemar's Test

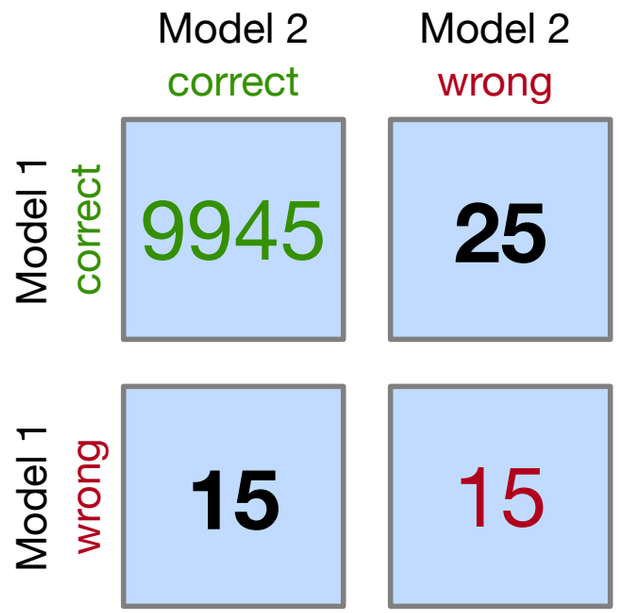
In both subpanel A and B, the accuracy of *Model 1* and *Model 2* are 99.7% and 99.6%, respectively.



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:

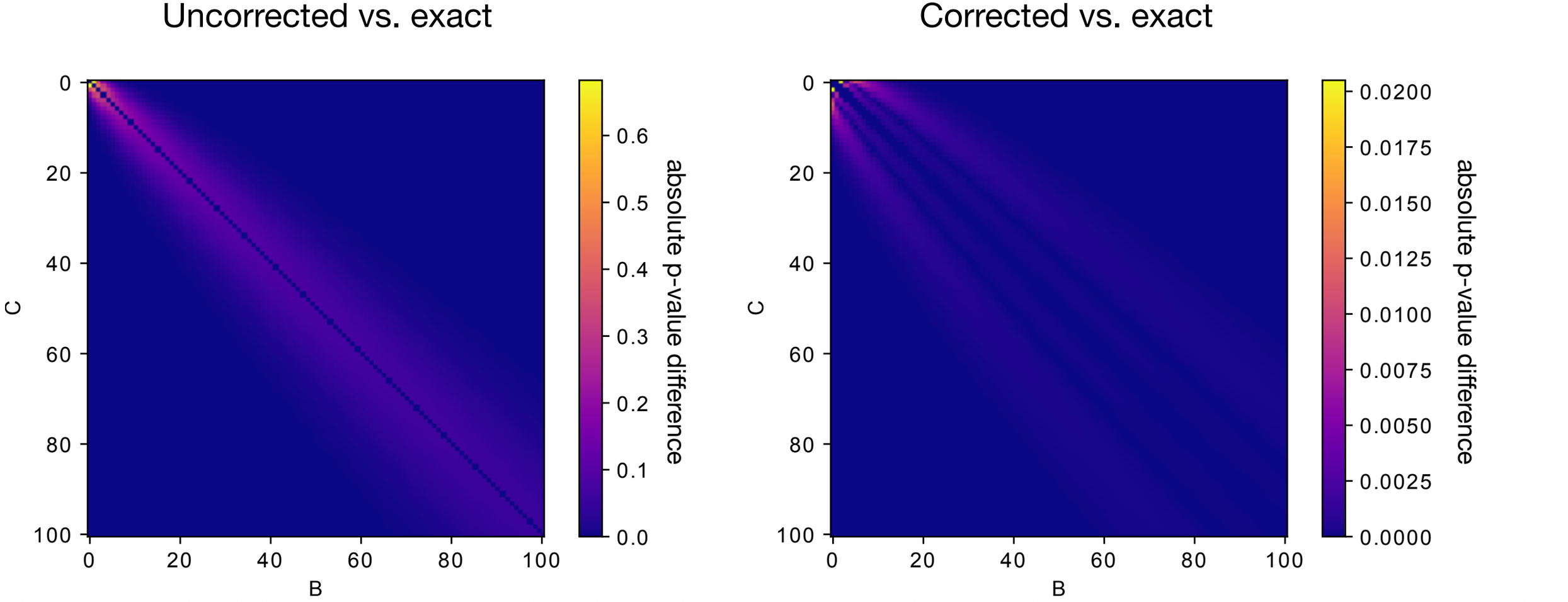
$$p = 2 \sum_{i=B}^{n} \binom{n}{i} 0.5^i (1 - 0.5)^{n-i},$$

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.)

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 M 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)

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

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

Cochran's Q Test

Let $\{C_1, \ldots, C_M\}$ be a set of classifiers who have all been tested on the same dataset. If the M classifiers don't perform differently, then the following Q statistic is distributed approximately as "chi-squared" with *M-1* degrees of freedom

$$Q = (M-1)\frac{M\sum_{i=1}^{M} G_i^2 - T^2}{MT - \sum_{j=1}^{n} M_j^2}$$

G_i is the number of objects out of *n* correctly classified by $C_i = 1,\ldots M$

M_j is the number of classifiers out of *M* that correctly classified the jth example in the test

and *T* is the total number of correct number of votes among the *M* classifiers

$$T = \sum_{i=1}^{M}; \quad G_i = \sum_{j=1}^{n} M_j$$

McNemar's Test with Bonferroni Correction to counteract the problem of multiple comparisons

Unfortunately, the problem of multiple comparisons receives little attention in literature. However, Peter H. Westfall, James F. Troendl, and Gene Pennello wrote a nice article on how to approach such situations where we want to compare multiple models to each other if you are interested:

- Westfall, Peter H., James F. Troendle, and Gene Pennello. "Multiple mcnemar tests." *Biometrics* 66.4 (2010): 1185-1191.

McNemar's Test with Bonferroni Correction to counteract the problem of multiple comparisons

Perneger, Thomas V. "What's wrong with Bonferroni adjustments." *BMJ: British Medical Journal* 316.7139 (1998): 1236:

> "Type I errors [False Positives] cannot decrease (the whole point of Bonferroni adjustments) without inflating type II errors (the probability of accepting the null hypothesis when the alternative is true) (Rothman, 1990). And type II errors [False Negatives] are no less false than type I errors."

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)

K-fold cross-validation with paired t test

$$t = \frac{\Delta ACC_{avg}\sqrt{k}}{\sqrt{\sum_{i=1}^{k} (\Delta ACC_i - \Delta ACC_{avg})^2/(k-1)}}$$

Here, k is the number of folds we use

$$\Delta ACC_{avg} = \frac{1}{k}\sum_{i=1}^{k} \Delta ACC_i \qquad \Delta ACC_i = ACC_i^A - ACC_i^B$$

H_0: equal accuracies

Resampled paired t test

$$t = \frac{\Delta ACC_{avg}\sqrt{k}}{\sqrt{\sum_{i=1}^{k} (\Delta ACC_i - \Delta ACC_{avg})^2/(k-1)}}$$

Here, k is the number of
times we split the set into train/test sets

$$\Delta ACC_{avg} = \frac{1}{k}\sum_{i=1}^{k} \Delta ACC_i \qquad \Delta ACC_i = ACC_i^A - ACC_i^B$$

H_0: equal accuracies

Two independence violations!!!

5x2 CV Cross-Validation + paired t test

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

Argument: independent training sets for 2-fold eval.

Now we get 2 differences, since we use 2-fold cross-validation:

$$\Delta ACC_i^{(1)} = ACC_i^{A(1)} - ACC_i^{B(1)}$$

$$\Delta ACC_i^{(2)} = ACC_i^{A(2)} - ACC_i^{B(2)}$$

$$\Delta ACC_{avg,i} = (\Delta ACC_i^{(1)} + \Delta ACC_i^{(2)})/2$$

est. variance: $s_i^2 = (ACC^{(1)} - \Delta ACC_{avg,i})^2 + (ACC^{(2)} - \Delta ACC_{avg,i})^2$

$$t = \frac{\Delta ACC_1^{(1)}}{\sqrt{(1/5)\sum_{i=1}^{5} s_i^2}}$$

(note that the subscript 1 in denominator is not a typo, it only refers to the first run)

Combined 5 × 2 cv F Test for Comparing Supervised Classification Learning Algorithms

Alpaydin, Ethem. "Combined 5×2 cv F test for comparing supervised classification learning algorithms." *Neural computation* 11.8 (1999): 1885-1892.

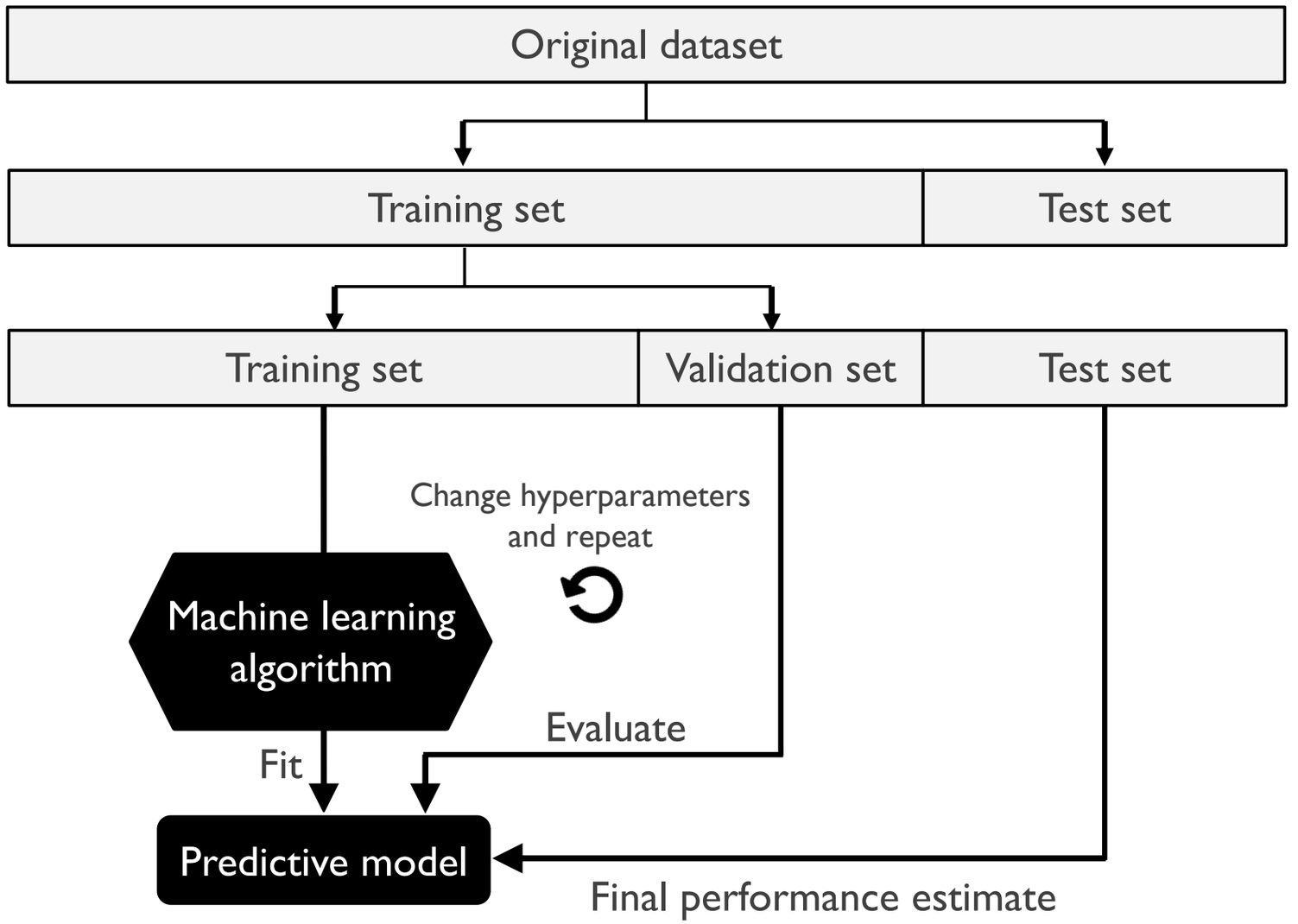
More robust than Dietterich 1998's 5x2 CV + t test

$$f = \frac{\sum_{i=1}^{5} \sum_{j=1}^{2} (\Delta ACC_i^j)^2}{2 \sum_{i=1}^{5} s_i^2}$$

Approximately F-distributed with 10 and 5 degrees of freedom.

Back to "Computational/Empirical" Methods

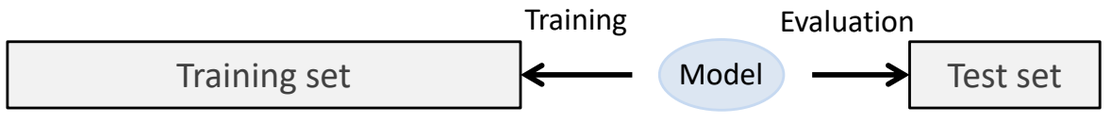
Recap: Model Selection with 3-way Holdout



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

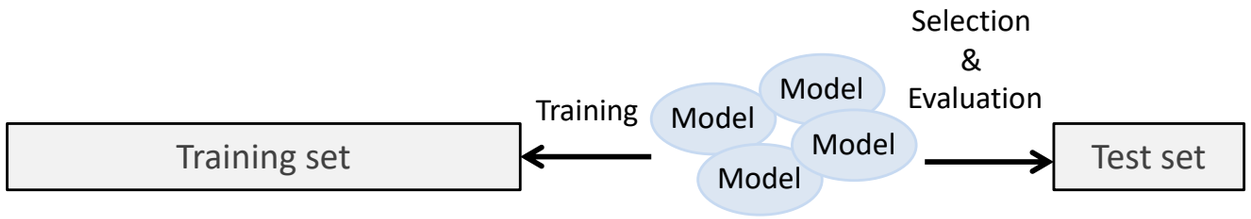
1)

good or **bad** ?



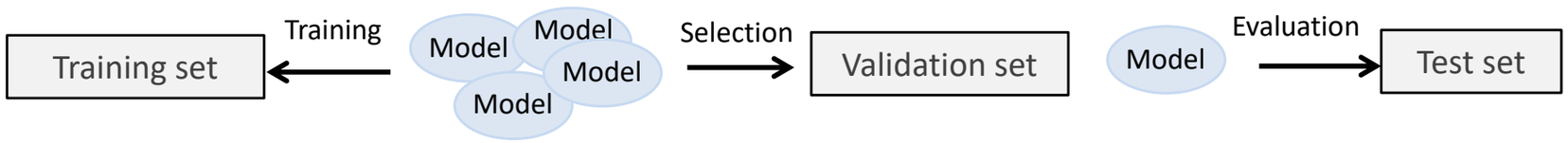
2)

good or **bad** ?



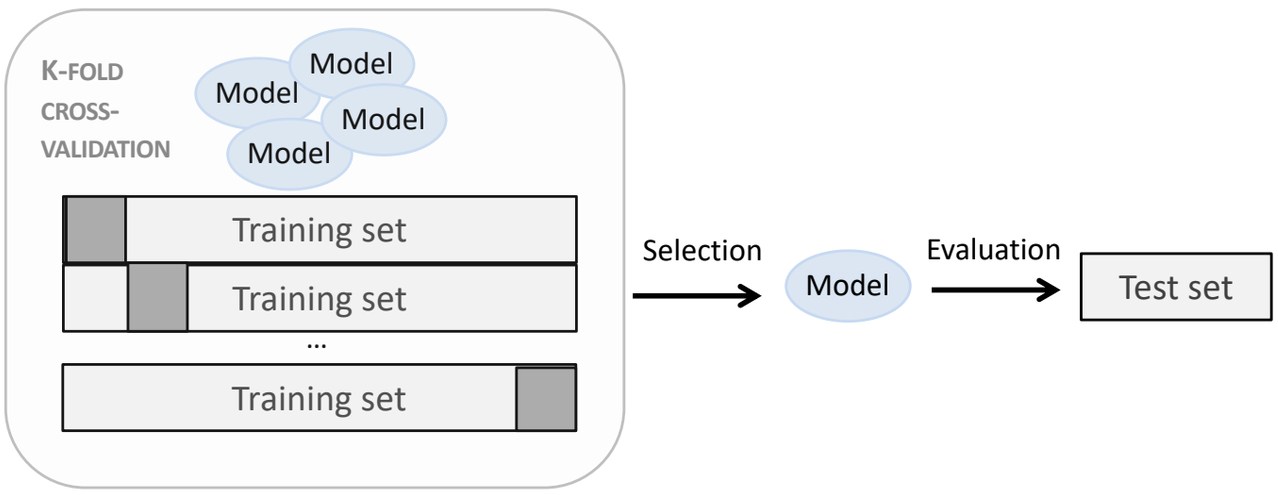
3)

good or **bad** ?



4)

good or **bad** ?

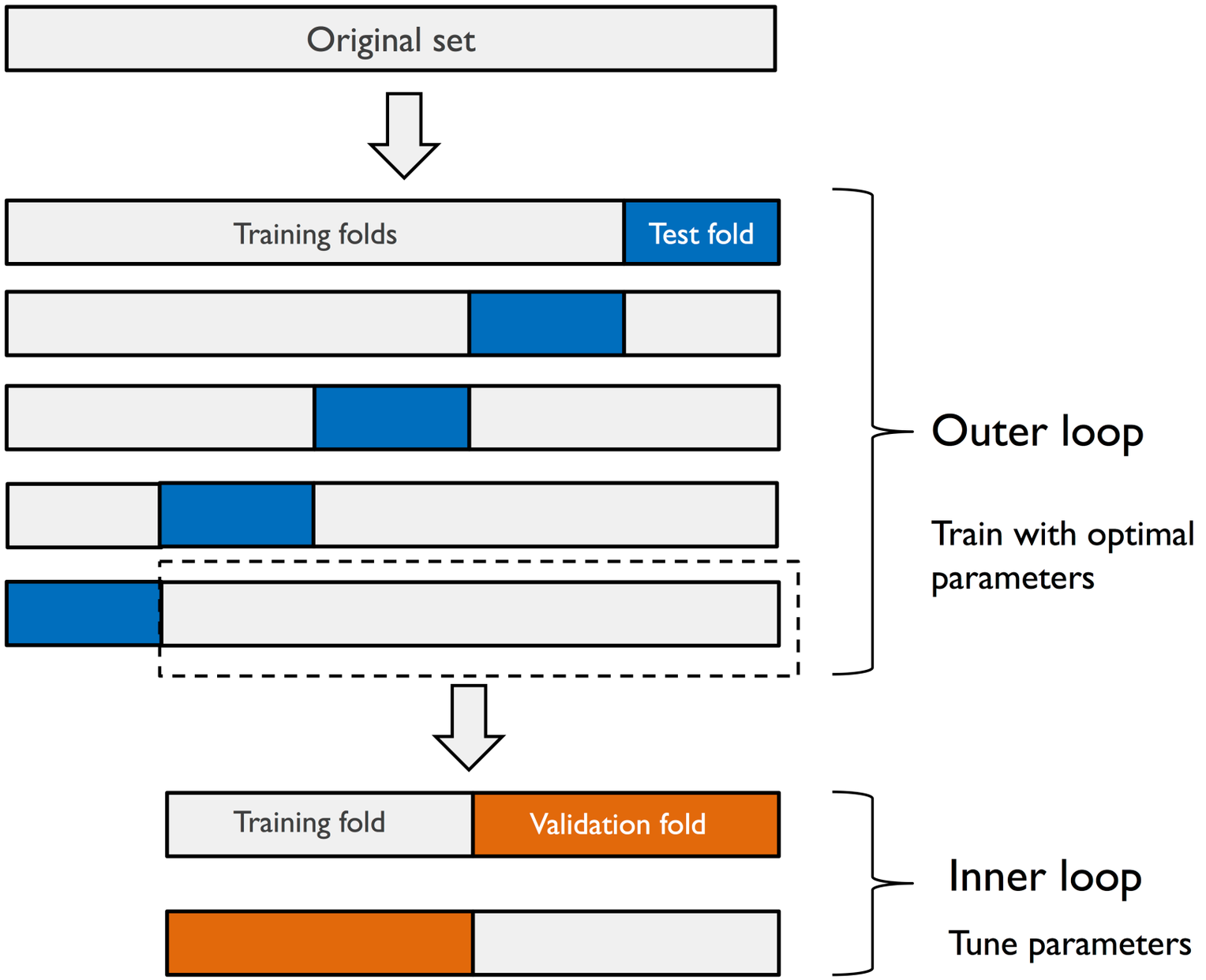


Nested Cross-Validation for Algorithm Selection

Main Idea:

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

Nested Cross-Validation

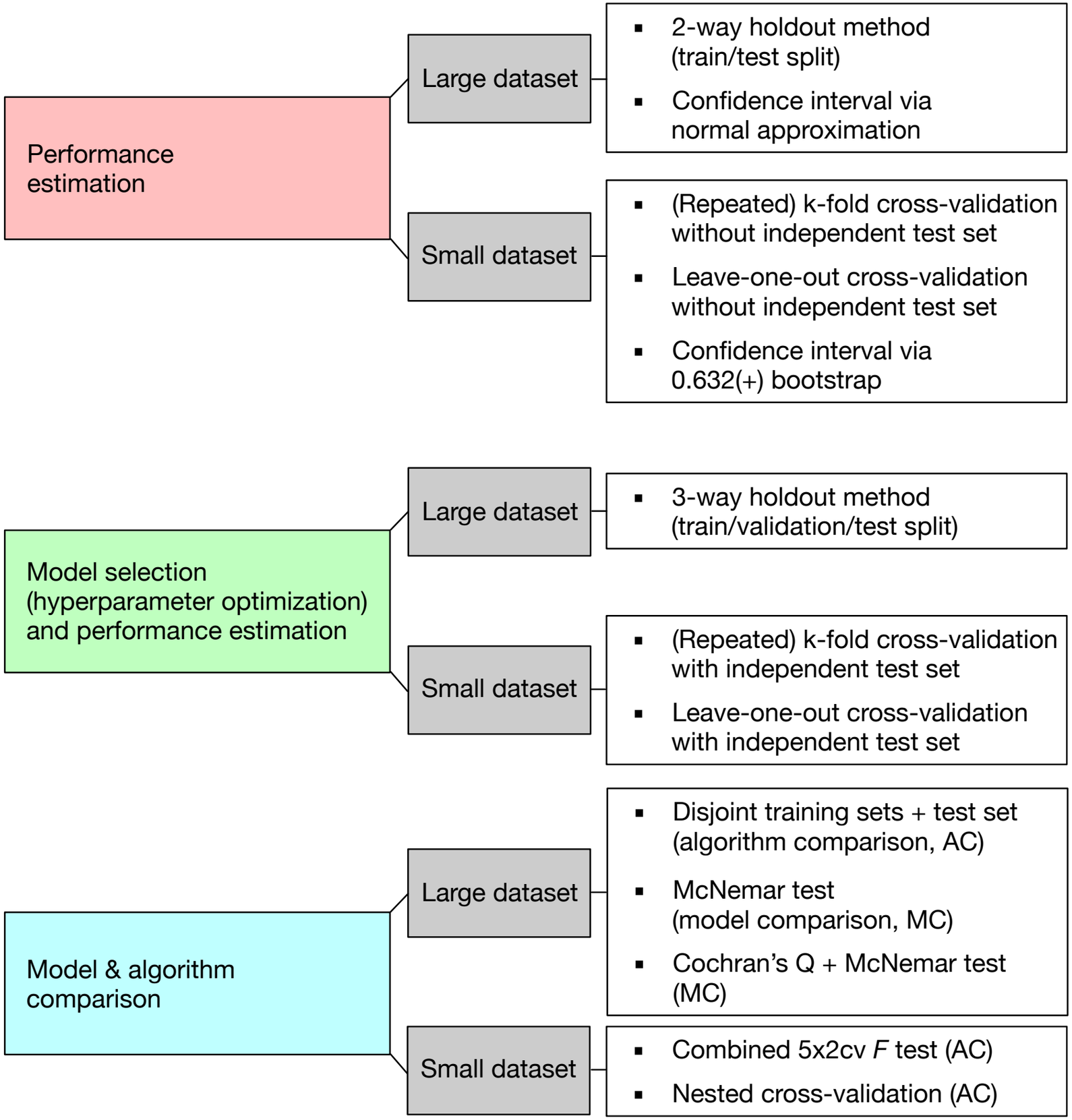


Nested Cross-Validation for Algorithm Selection

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

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

Conclusions, (my) "recommendations"



Reading Assignments

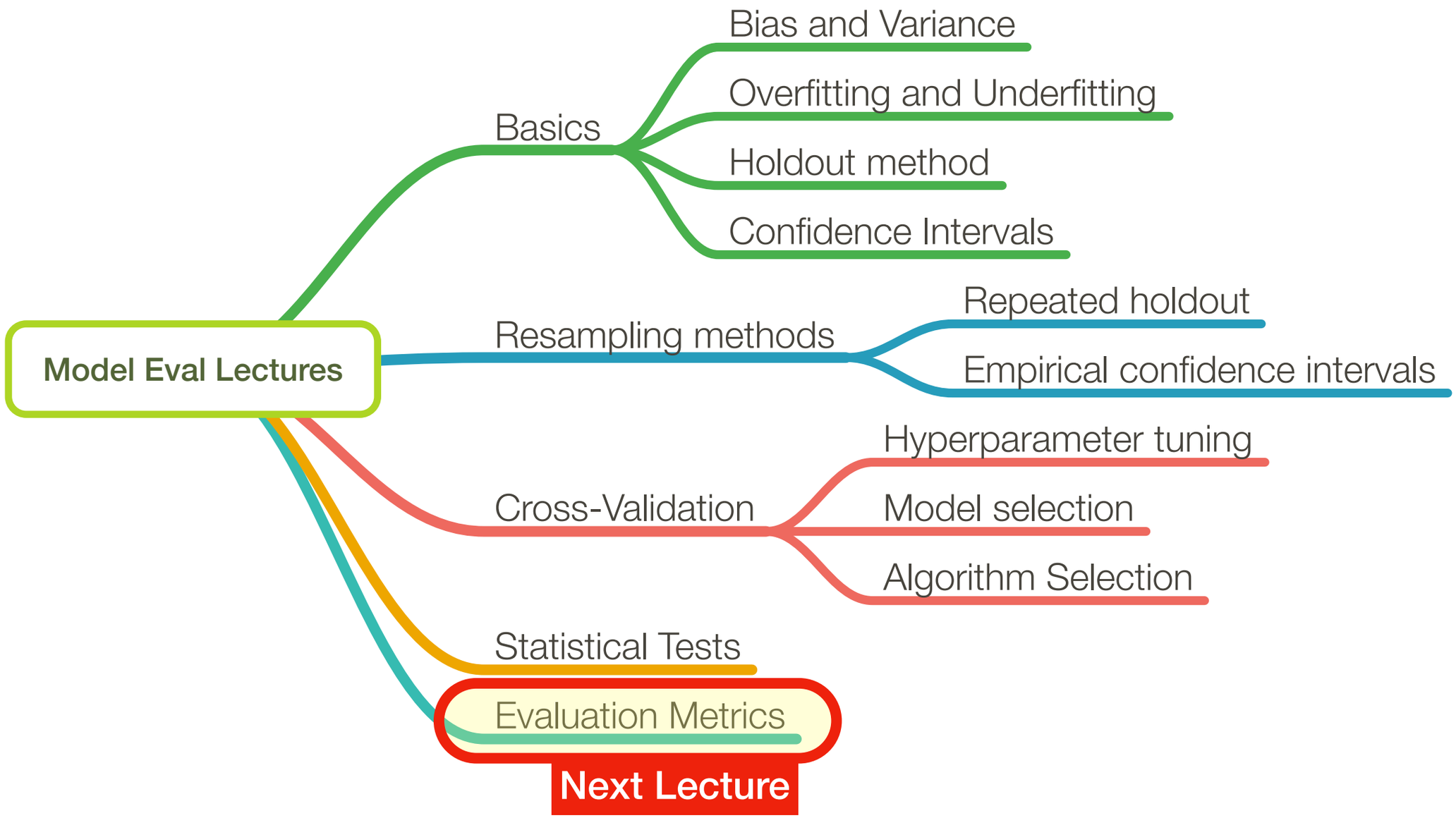
L11 Lecture notes:

https://github.com/rasbt/stat479-machine-learning-fs19/blob/master/11_eval4-algo/11-eval4-algo__notes.pdf

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/
- F-Test http://rasbt.github.io/mlxtend/user_guide/evaluate/ftest/
- 5x2cv combined *F* test http://rasbt.github.io/mlxtend/user_guide/evaluate/combined_ftest_5x2cv/
- Nested Cross-Validation https://github.com/rasbt/stat479-machine-learning-fs19/blob/master/11_eval-algo/11_eval-algo_code.ipynb

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