Lecture 10

Model Evaluation 3: Cross Validation

STAT 451: Machine Learning, Fall 2020

Sebastian Raschka

http://stat.wisc.edu/~sraschka/teaching/stat451-fs2020/

1. Lecture Overview

2. Hyperparameters

3. Cross-validation for model evaluation

4. CV for model evaluation code examples

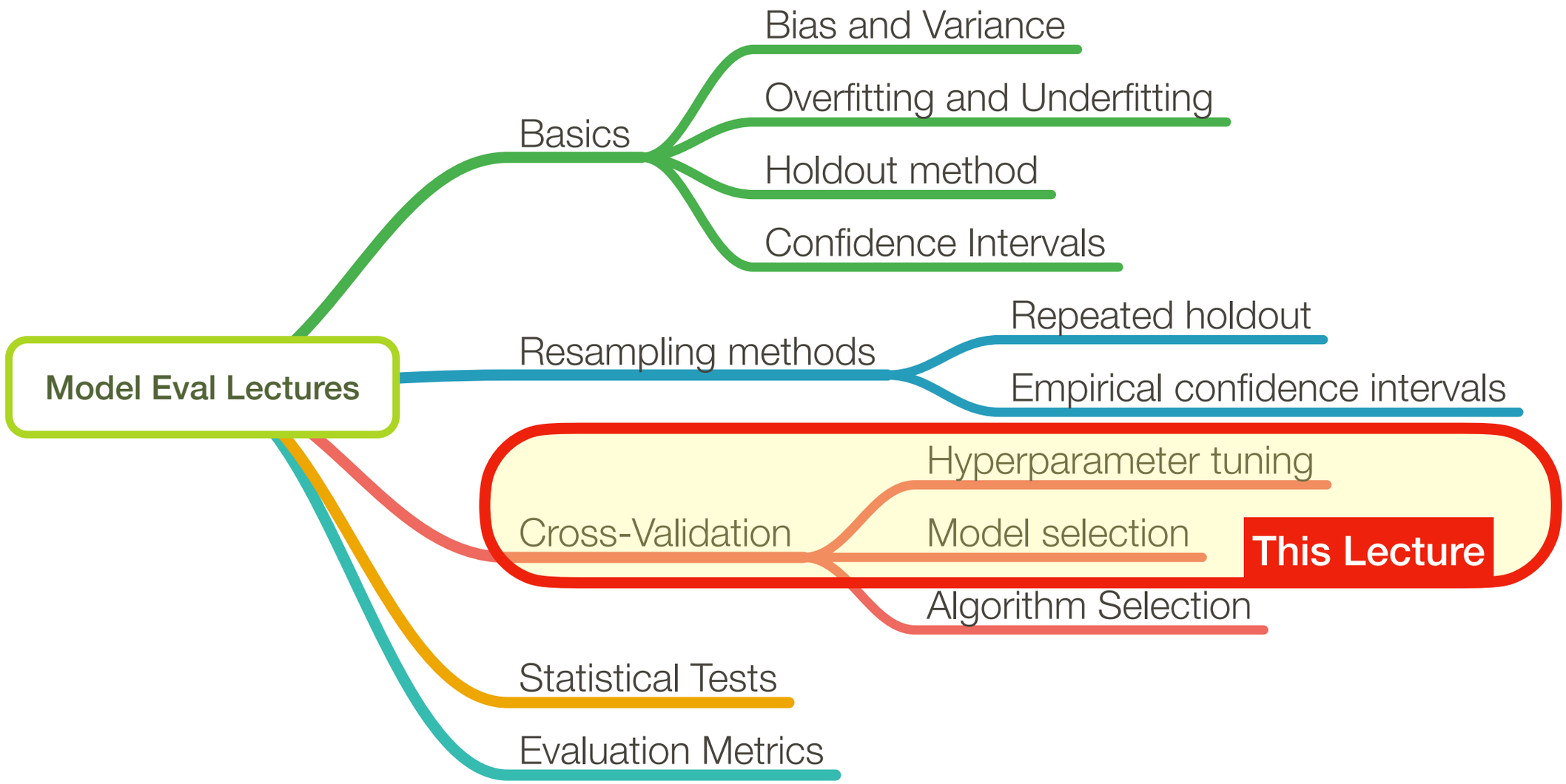
5. Cross-validation for model selection

6. CV for model selection code examples

7. The 1-standard error method

8. 1std err. code examples

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

- **Performance estimation**
  - Large dataset
    - 2-way holdout method (train/test split)
    - Confidence interval via normal approximation
  - Small dataset
    - (Repeated) k-fold cross-validation without independent test set
    - Leave-one-out cross-validation without independent test set
    - Confidence interval via 0.632(+) bootstrap
- **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)

These are my personal recommendations;

MC = model comparison,
AC = algorithm comparison;

Terms that are still unfamiliar (McNemar's test, 5x2cv F test, etc.) will be covered next lecture.

1. Lecture Overview

2. Hyperparameters

3. Cross-validation for model evaluation

4. CV for model evaluation code examples

5. Cross-validation for model selection

6. CV for model selection code examples

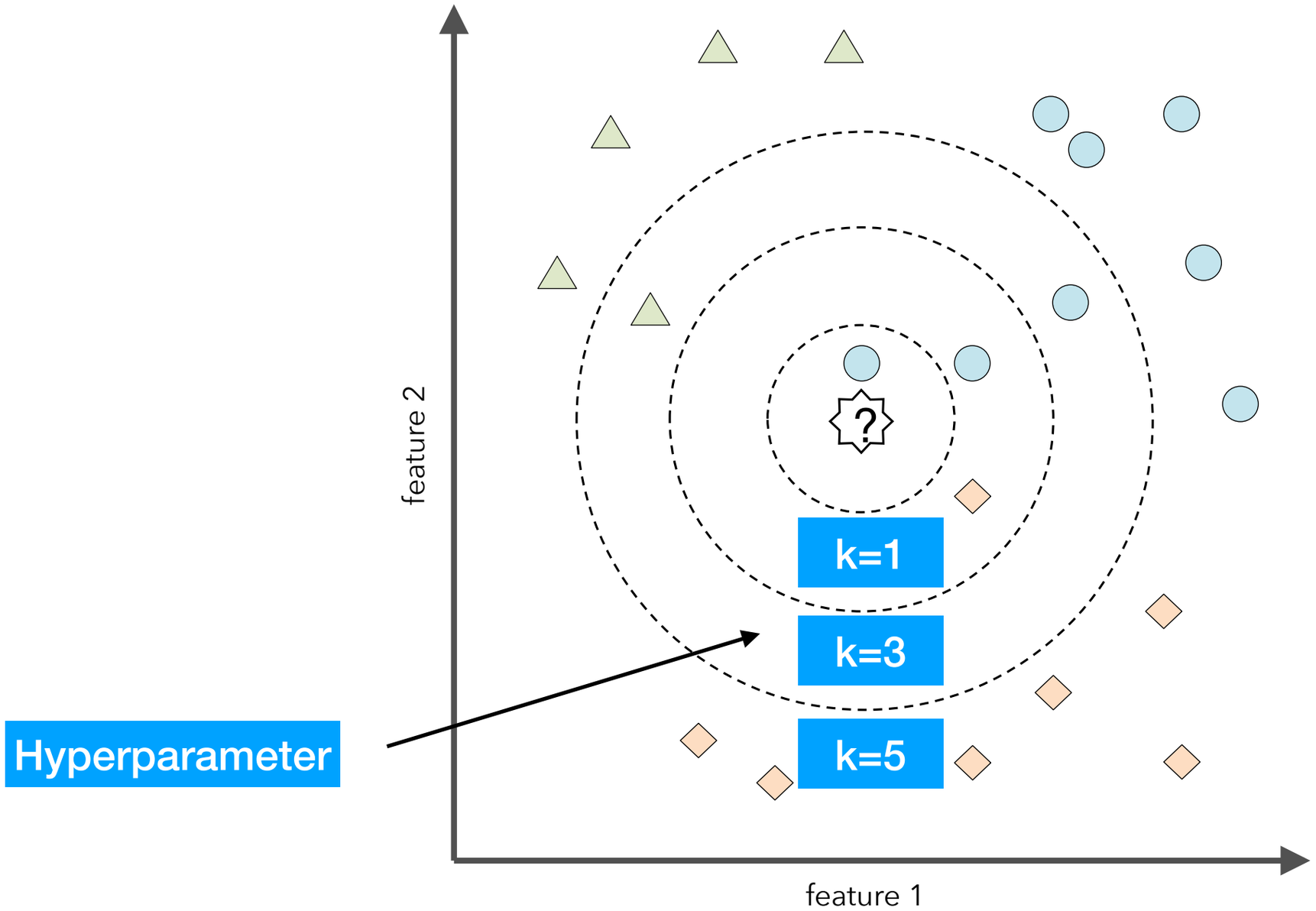
7. The 1-standard error method

8. 1std err. code examples

What are Hyperparameters?

Hyperparameters

nonparametric model: k-nearest neighbors

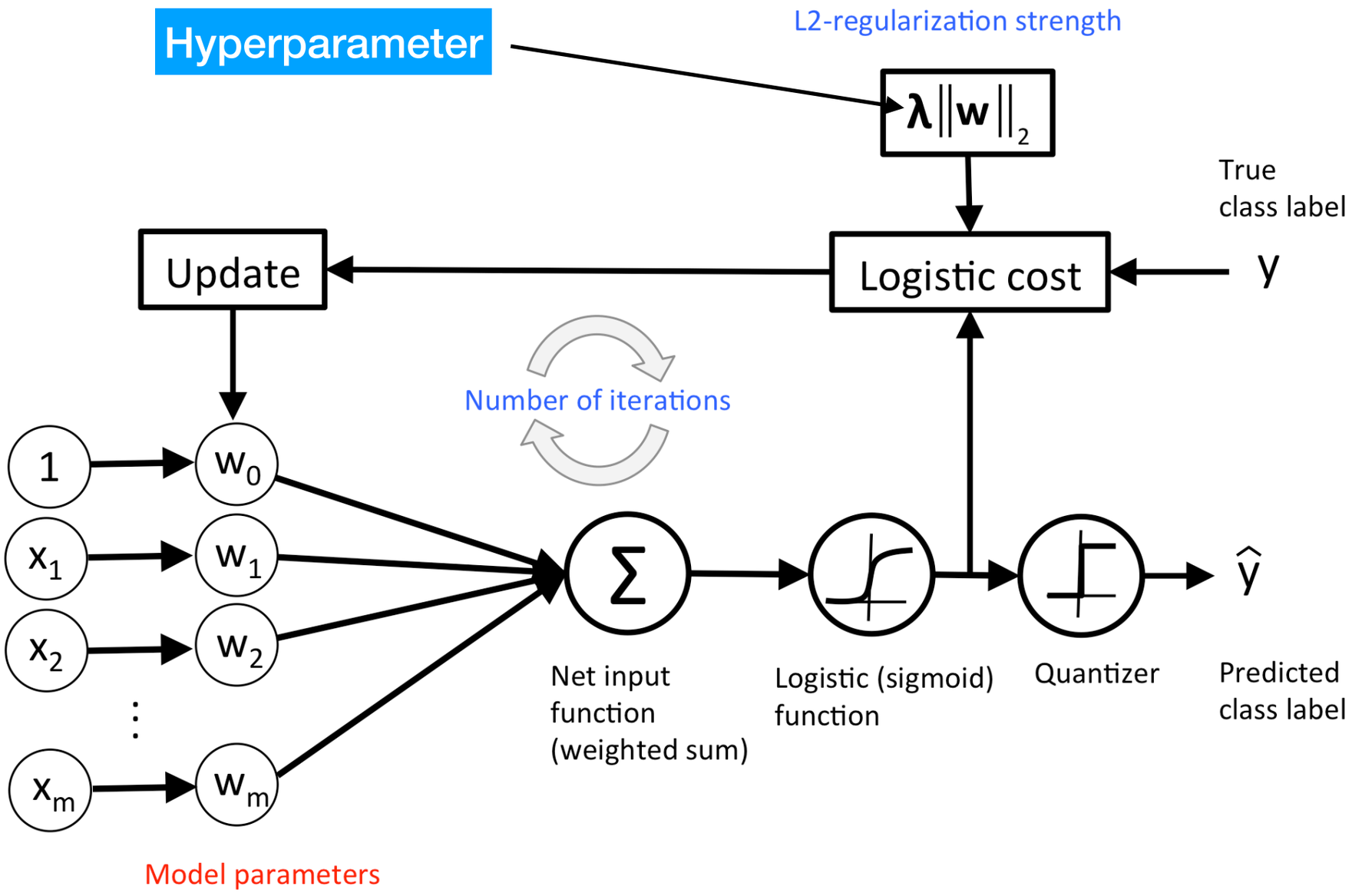


class sklearn.tree.**DecisionTreeClassifier**(*, *criterion='gini'*, *splitter='best'*, *max_depth=None*, *min_samples_split=2*, *min_samples_leaf=1*, *min_weight_fraction_leaf=0.0*, *max_features=None*, *random_state=None*, *max_leaf_nodes=None*, *min_impurity_decrease=0.0*, *min_impurity_split=None*, *class_weight=None*, *presort='deprecated'*, *ccp_alpha=0.0*)

class sklearn.ensemble.**HistGradientBoostingClassifier**(*loss='auto'*, *, *learning_rate=0.1*, *max_iter=100*, *max_leaf_nodes=31*, *max_depth=None*, *min_samples_leaf=20*, *l2_regularization=0.0*, *max_bins=255*, *monotonic_cst=None*, *warm_start=False*, *early_stopping='auto'*, *scoring='loss'*, *validation_fraction=0.1*, *n_iter_no_change=10*, *tol=1e-07*, *verbose=0*, *random_state=None*)

Hyperparameters

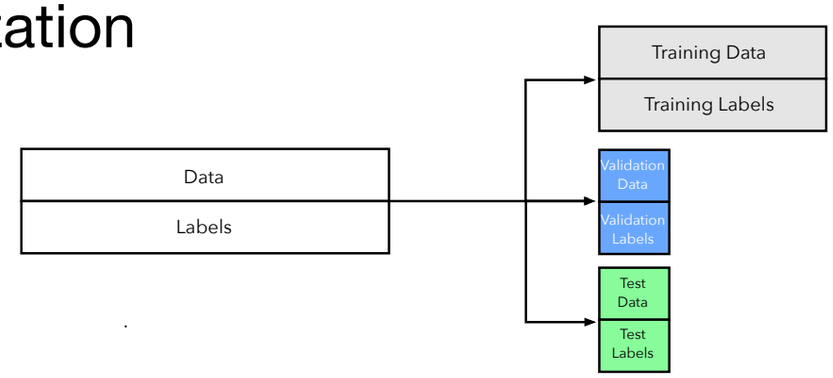
parametric model: logistic regression



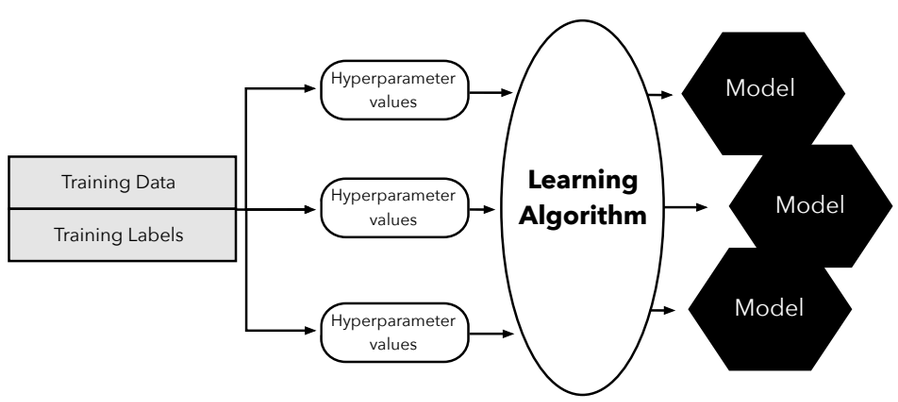
3-Way Holdout

instead of "regular" holdout to avoid "data leakage" during hyperparameter optimization

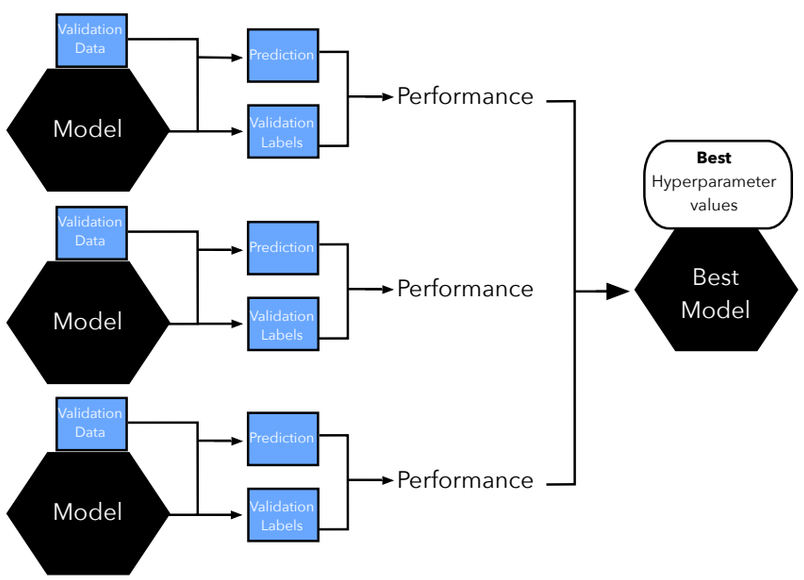
1



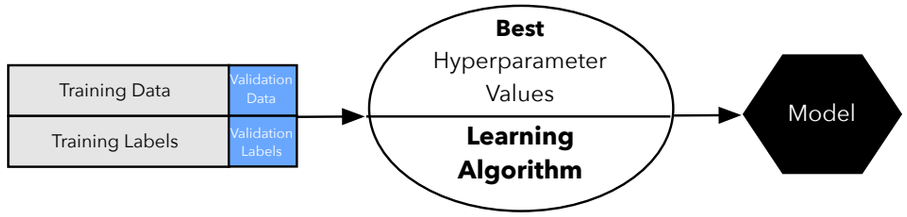
2



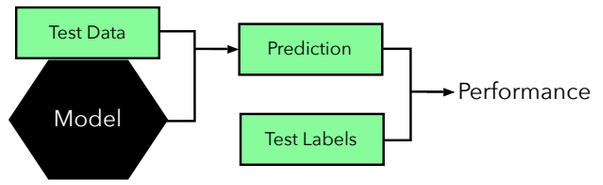
3



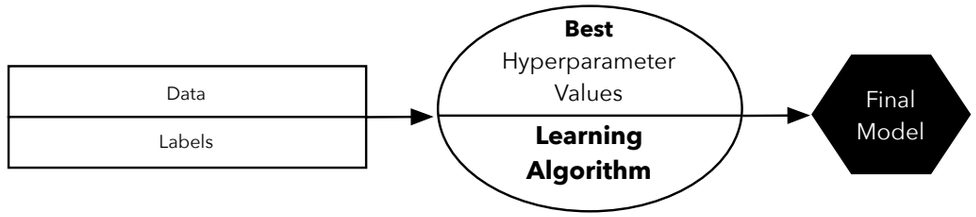
4



5



6



Main points why we evaluate the predictive performance of a model:

1. Want to estimate the generalization performance, the predictive performance of our model on future (unseen) data.

2. Want to increase the predictive performance by tweaking the learning algorithm and selecting the best performing model from a given hypothesis space.

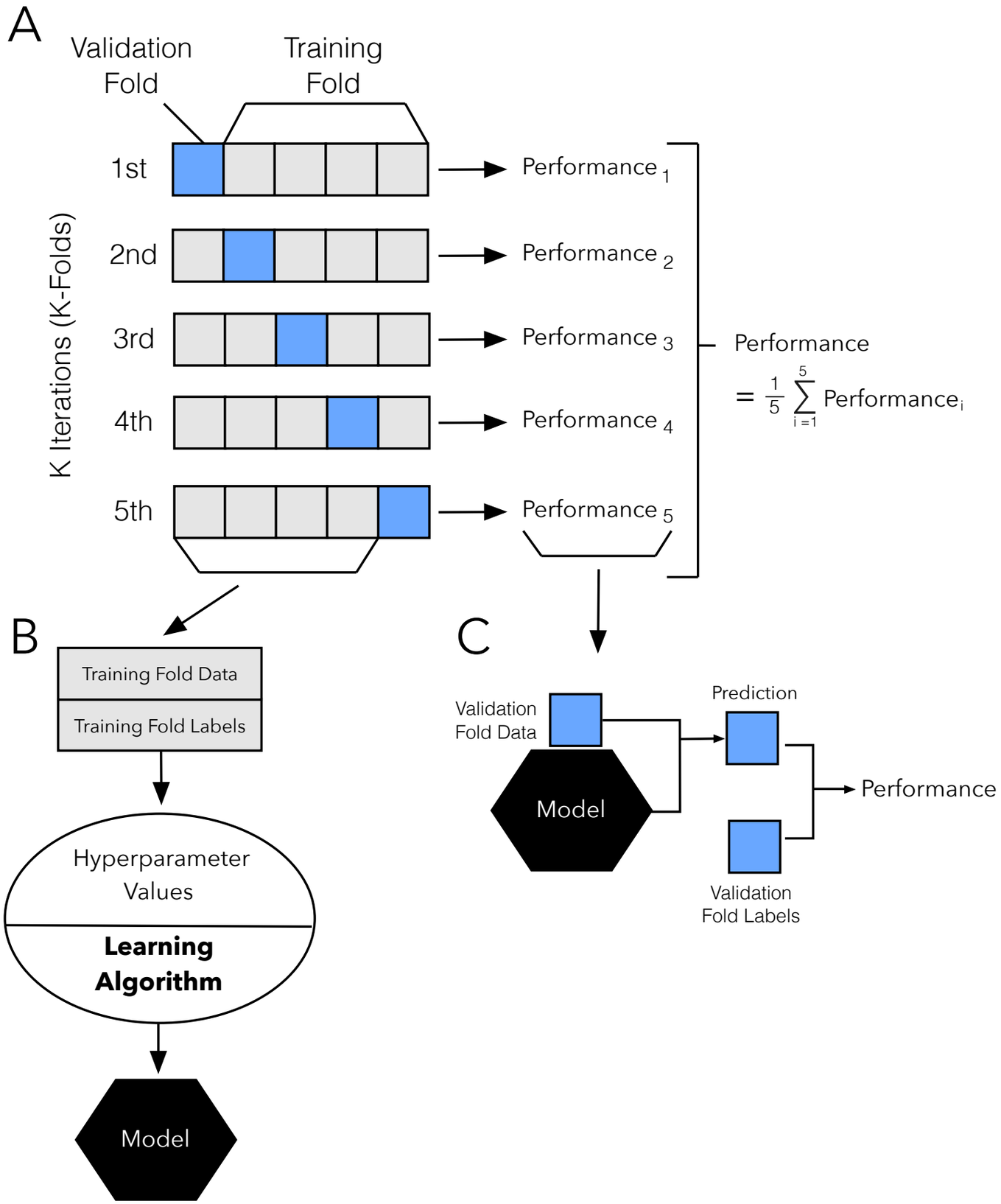
3. Want to identify the ML algorithm that is best-suited for the problem at hand; thus, we want to compare different algorithms, selecting the best-performing one as well as the best performing model from the algorithm's hypothesis space.

1. Lecture Overview
2. Hyperparameters
3. **Cross-validation for model evaluation**
4. CV for model evaluation code examples
5. Cross-validation for model selection
6. CV for model selection code examples
7. The 1-standard error method
8. 1std err. code examples

k-Fold Cross-Validation Part 1

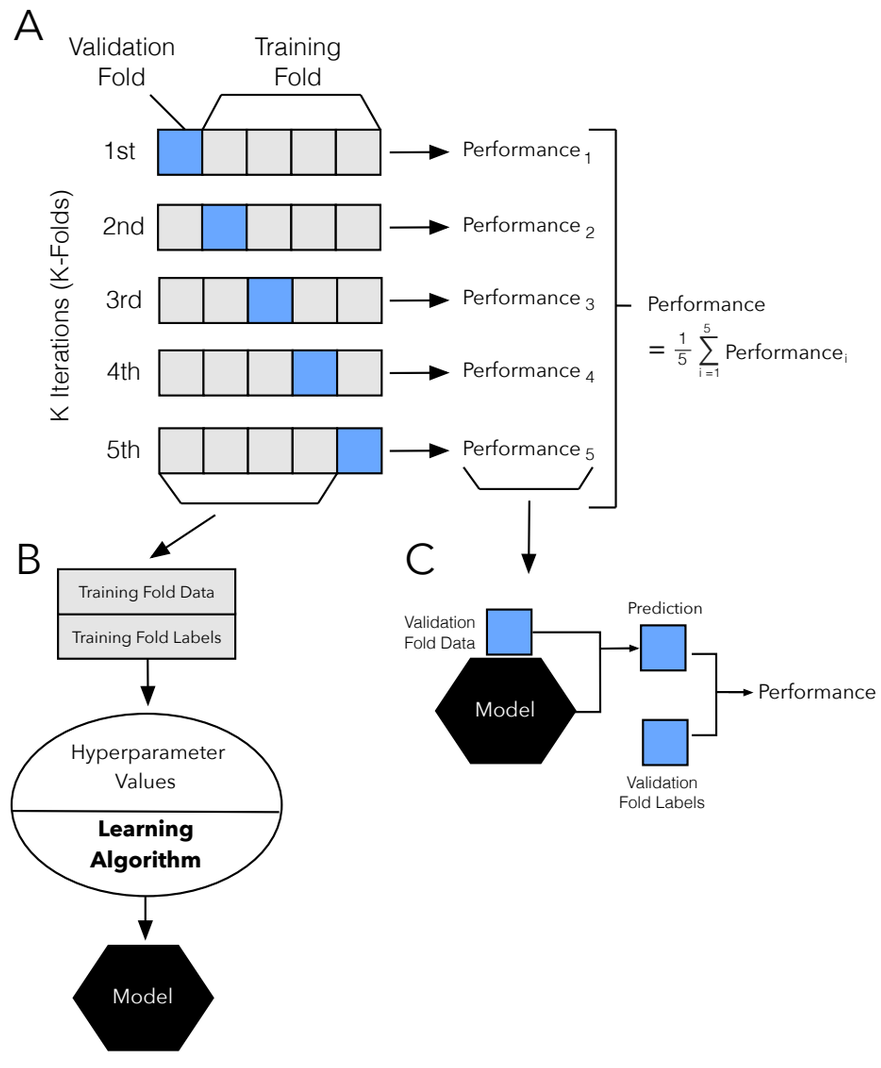
Model Evaluation

k-Fold Cross-Validation

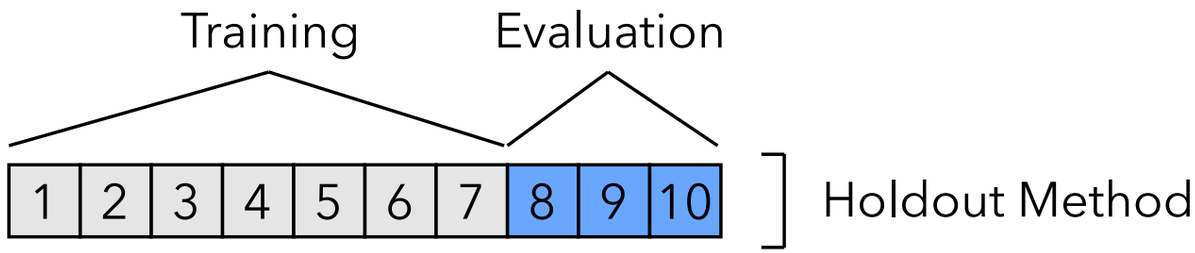


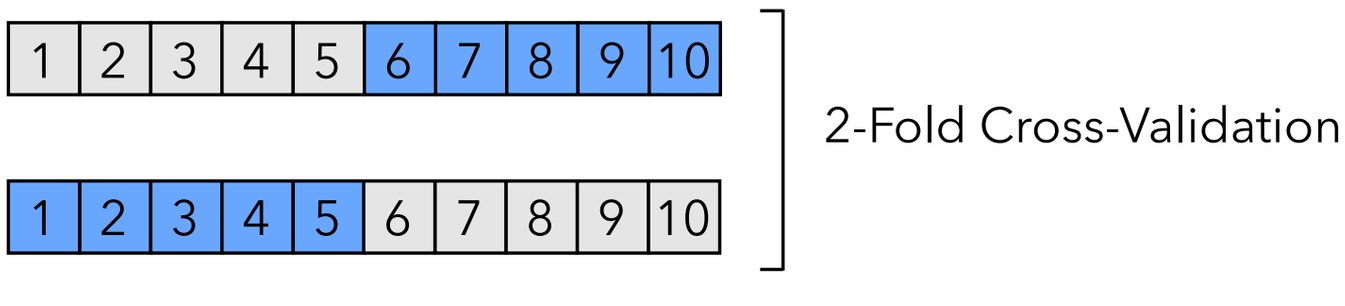
k-Fold Cross-Validation

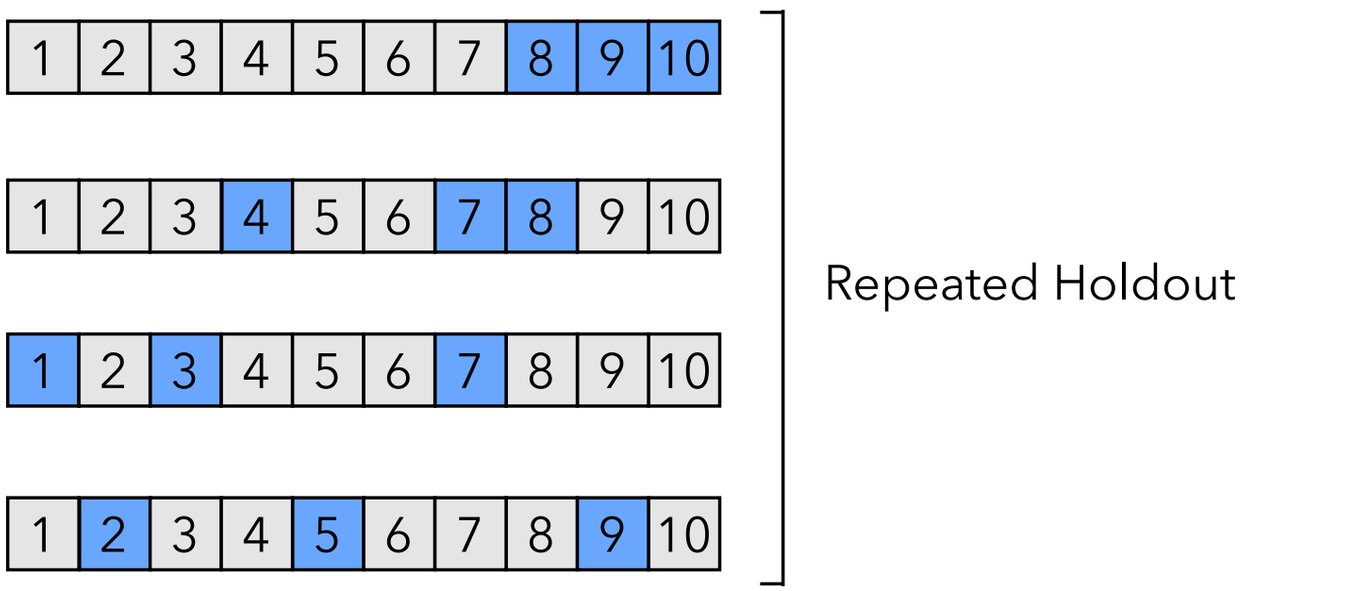
- non-overlapping validation folds; utilizes all data for testing
- overlapping training folds
- some variance estimate from different training sets, (but no unbiased estimate)
- more pessimistic for small k because we withhold data from fitting



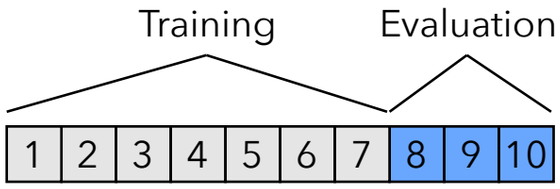
k-Fold CV special cases: k=2 & k=n



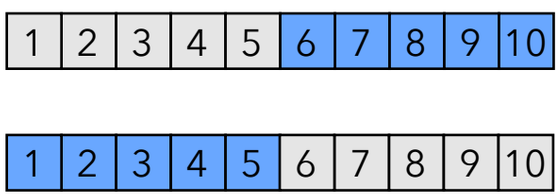




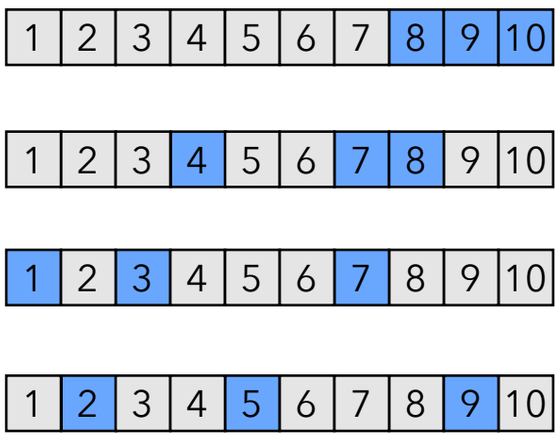
k-Fold CV special cases: k=2 & k=n



Holdout Method



2-Fold Cross-Validation



Repeated Holdout

LOOCV

k-Fold Cross-Validation

"[...] where available sample sizes are modest, holding back compounds for model testing is ill-advised. This fragmentation of the sample harms the calibration and does not give a trustworthy assessment of fit anyway. It is better to use all data for the calibration step and check the fit by cross-validation, making sure that the cross-validation is carried out correctly. [...] The only motivation to rely on the holdout sample rather than cross-validation would be if there was reason to think the cross-validation not trustworthy -- biased or highly variable. But neither theoretical results nor the empiric results sketched here give any reason to disbelieve the cross-validation results." [1]

1. Hawkins, D. M., Basak, S. C., & Mills, D. (2003). Assessing model fit by cross-validation. *Journal of chemical information and computer sciences*, *43*(2), 579-586.

LOOCV vs Holdout

Experiment	**Mean**	**Standard deviation**
True R^2 — q^2	0.010	0.149
True R^2 — hold 50	0.028	0.184
True R^2 — hold 20	0.055	0.305
True R^2 — hold 10	0.123	0.504

1. Hawkins, D. M., Basak, S. C., & Mills, D. (2003). Assessing model fit by cross-validation. *Journal of Chemical Information and Computer Sciences*, *43*(2), 579-586.

The reported "mean" refers to the averaged difference between the true coefficients of determination (R^2) and the coefficients obtained via LOOCV (here called q^2) after repeating this procedure on multiple, different 100-example training sets

In rows 2-4, the researchers used the holdout method for fitting models to the 100-example training sets, and they evaluated the performances on holdout sets of sizes 10, 20, and 50 samples. Each experiment was repeated 75 times, and the mean column shows the average difference between the estimated R^2 and the true R^2 values.

(why not changing the random seed in LOOCV?)

Problems with LOOCV for Classification

- While LOOCV is almost unbiased, one downside of using LOOCV over k-fold cross-validation with $k < n$ is the large variance of the LOOCV estimate.

- LOOCV is "defect" when using a discontinuous loss-function such as the 0-1 loss in classification or even in continuous loss functions such as the mean-squared-error.

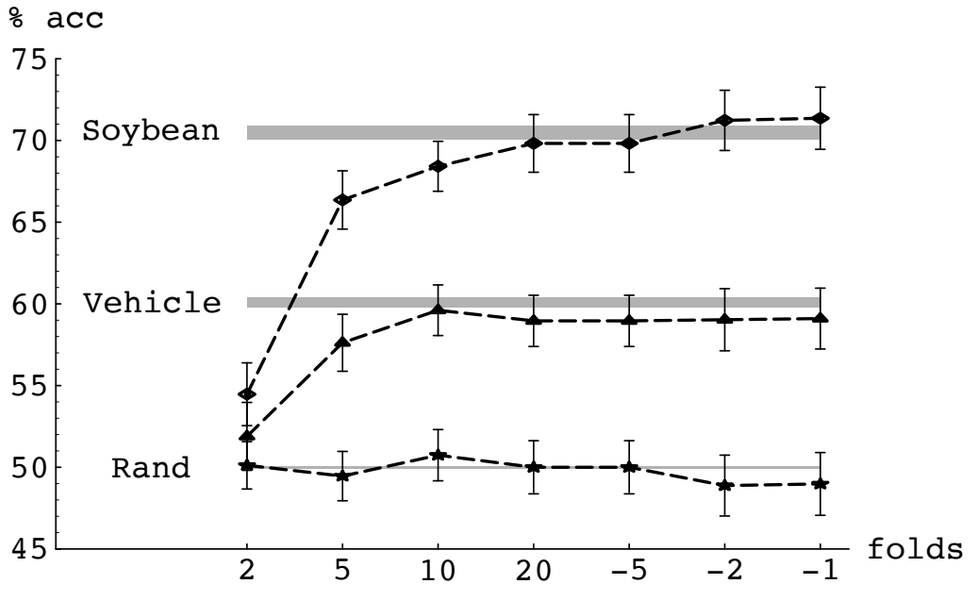
- LOOCV has high variance because the test set only contains one example.

Problems with LOOCV for Classification

"With *k=n*, the cross-validation estimator is approximately unbiased for the true (expected) prediction error, but can have high variance because the n "training sets" are so similar to one another." [1]

[1] Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* (Vol. 1, No. 10). New York, NY, USA: Springer series in statistics.

Empirical Study and Recommendation



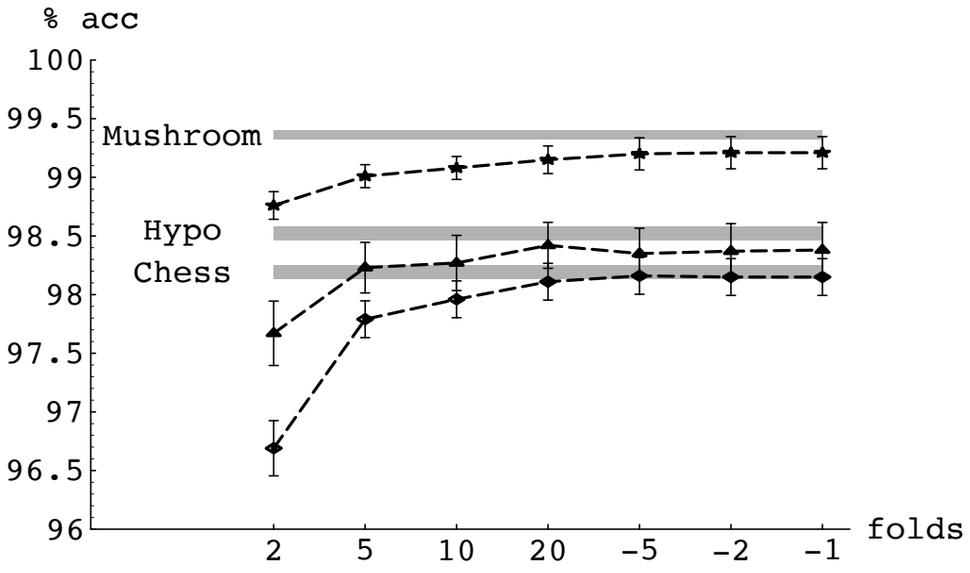
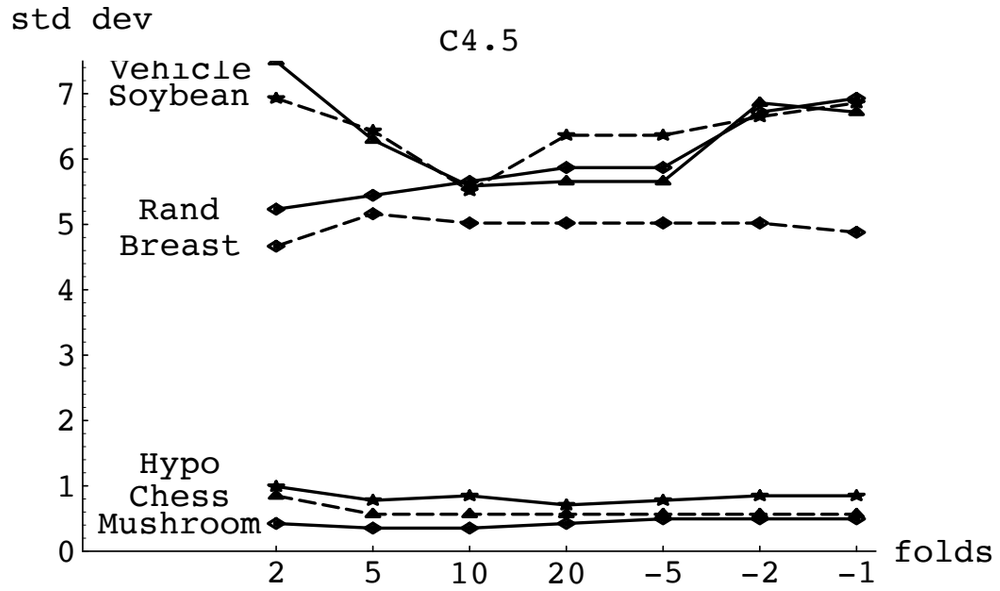


Figure 1: C4.5: The bias of cross-validation with varying folds. A negative k folds stands for leave-k-out. Error bars are 95% confidence intervals for the mean. The gray regions indicate 95% confidence intervals for the true accuracies. Note the different ranges for the accuracy axis.



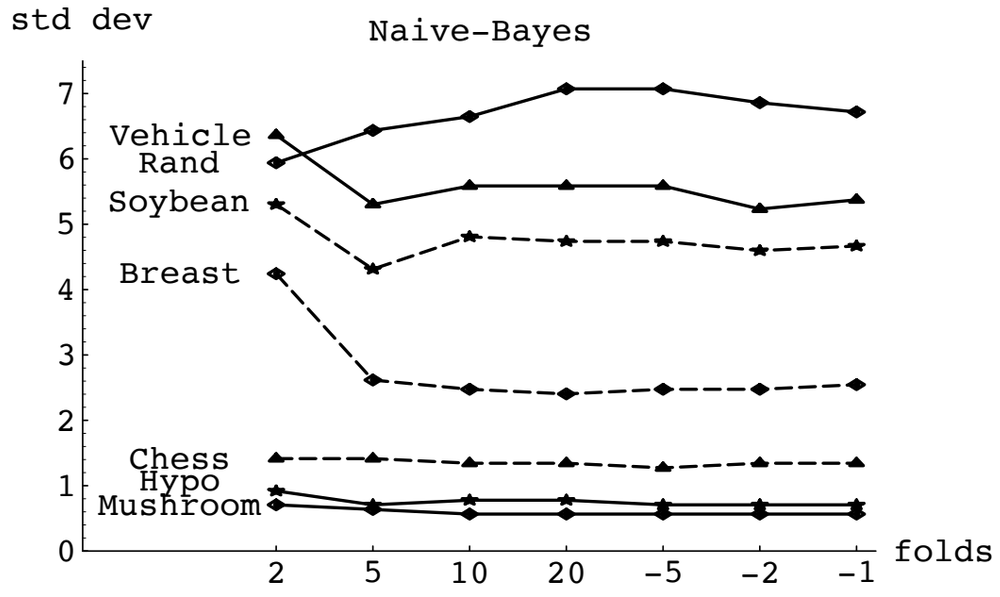


Figure 3: Cross-validation: standard deviation of accuracy (population). Different line styles are used to help differentiate between curves.

Kohavi, R. (1995, August). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai* (Vol. 14, No. 2, pp. 1137-1145).

Summarizing k-Fold CV for Model Evaluation

What happens if we increase *k*?

- The bias of the performance estimator ____creases (more accurate / more variable?)
- The variance of the performance estimators ____creases (more accurate / more variable?)
- The computational cost ____creases (more iterations, larger training sets during fitting)
- Exception: decreasing the value of k in k-fold cross-validation to small values (for example, 2 or 3) also ____creases the variance on small datasets due to random sampling effects

1. Lecture Overview
2. Hyperparameters
3. Cross-validation for model evaluation
4. **CV for model evaluation code examples**
5. Cross-validation for model selection
6. CV for model selection code examples
7. The 1-standard error method
8. 1std err. code examples

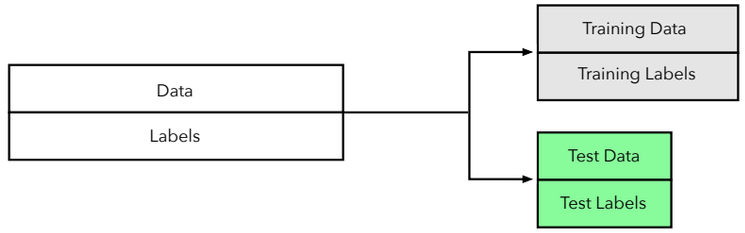
https://github.com/rasbt/stat451-machine-learning-fs20/blob/master/L10/code/10_04_kfold-eval.ipynb

1. Lecture Overview
2. Hyperparameters
3. Cross-validation for model evaluation
4. CV for model evaluation code examples
5. **Cross-validation for model selection**
6. CV for model selection code examples
7. The 1-standard error method
8. 1std err. code examples

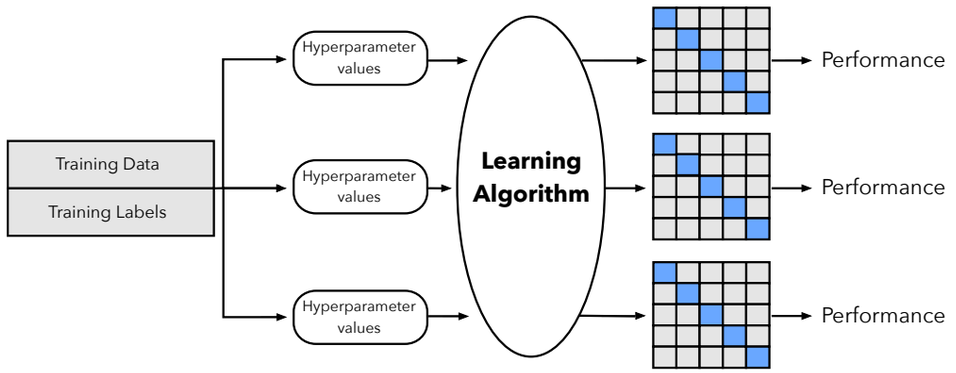
k-Fold Cross-Validation Part 2

Model Selection

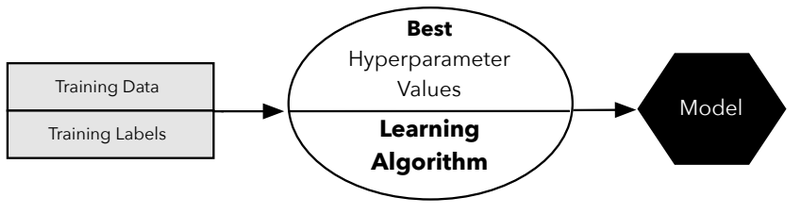
1


Data
Labels
Training Data
Training Labels
Test Data
Test Labels

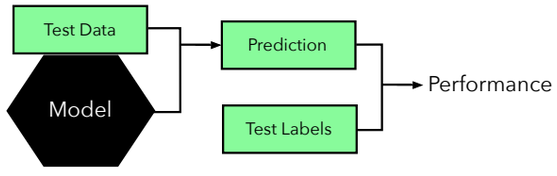
2


Training Data
Training Labels
Hyperparameter values
Hyperparameter values
Hyperparameter values
Learning Algorithm
Performance
Performance
Performance

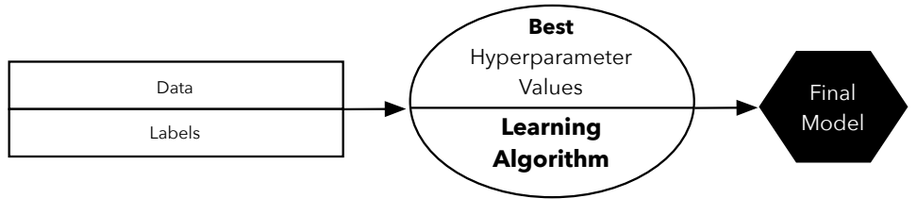
3


Training Data
Training Labels
Best
Hyperparameter
Values
Learning
Algorithm
Model

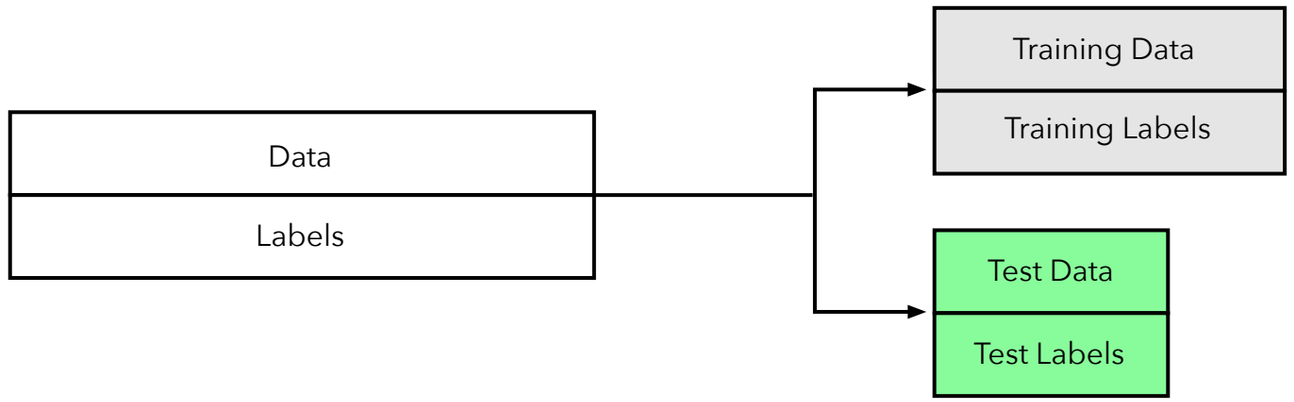
4


Test Data
Model
Prediction
Test Labels
Performance

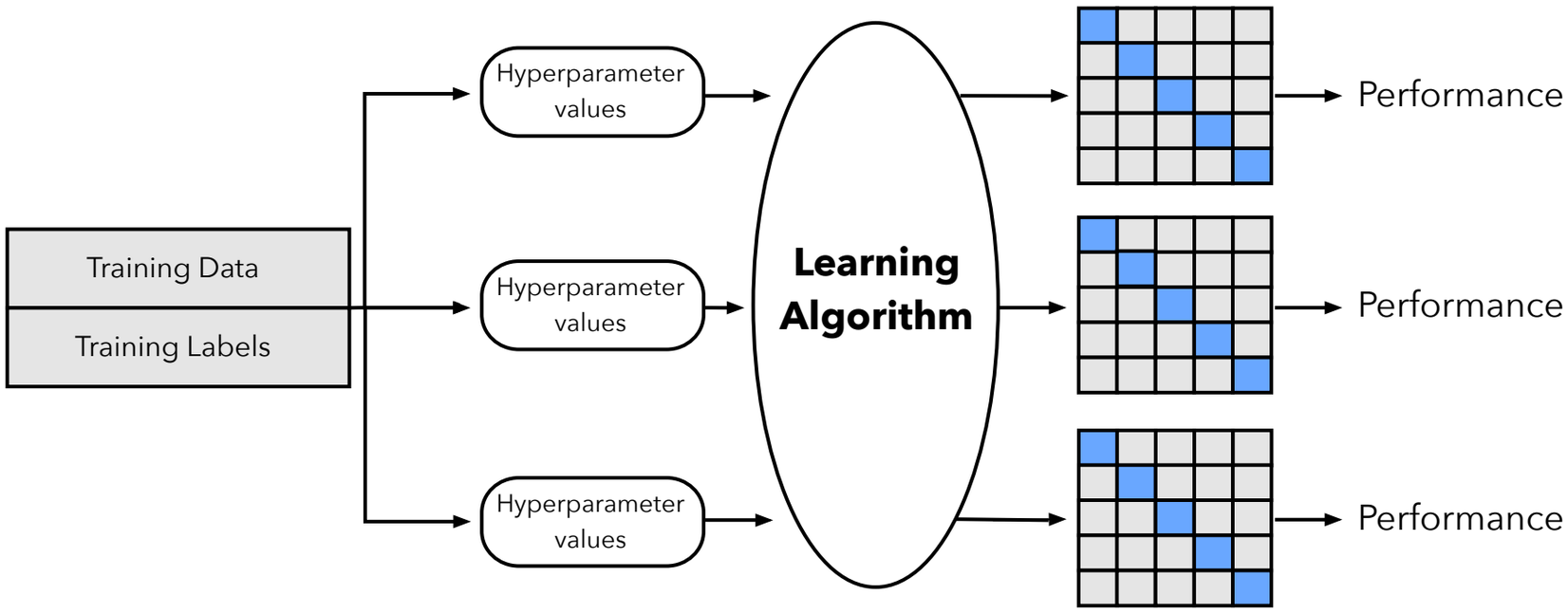
5


Data
Labels
Best
Hyperparameter
Values
Learning
Algorithm
Final
Model

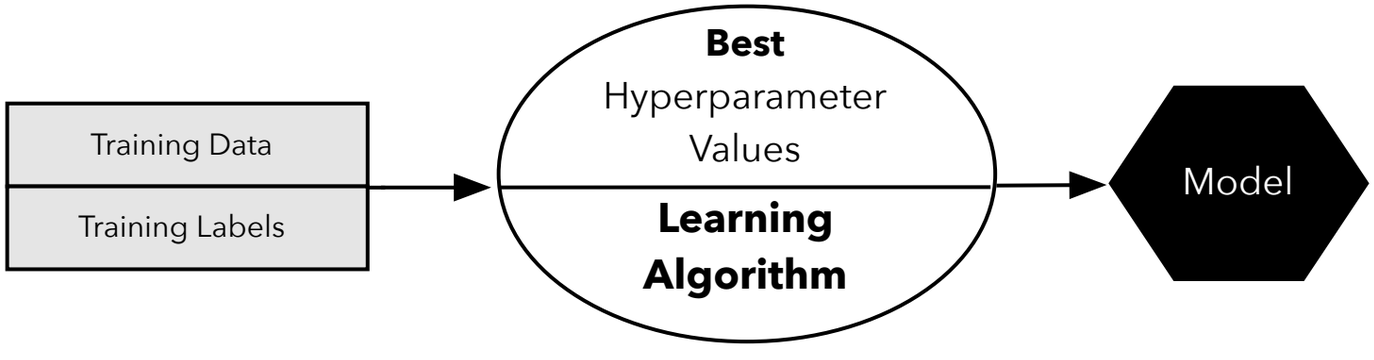
1


Data
Labels
Training Data
Training Labels
Test Data
Test Labels

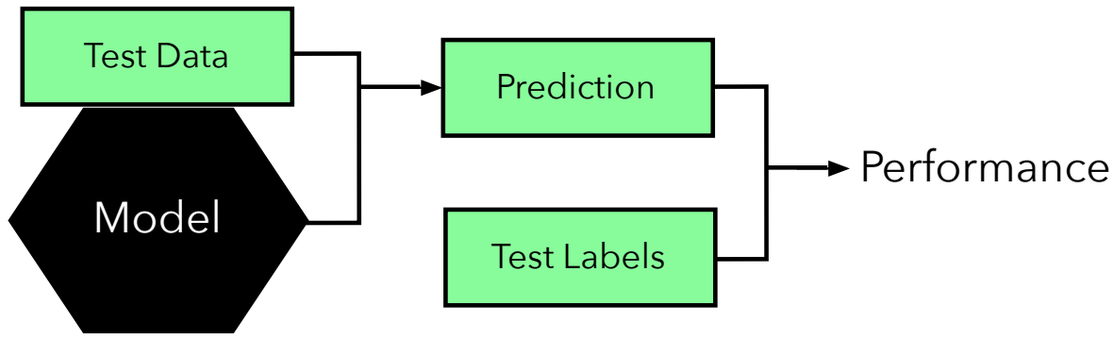
2


Training Data
Training Labels
Hyperparameter values
Hyperparameter values
Hyperparameter values
Learning Algorithm
Performance
Performance
Performance

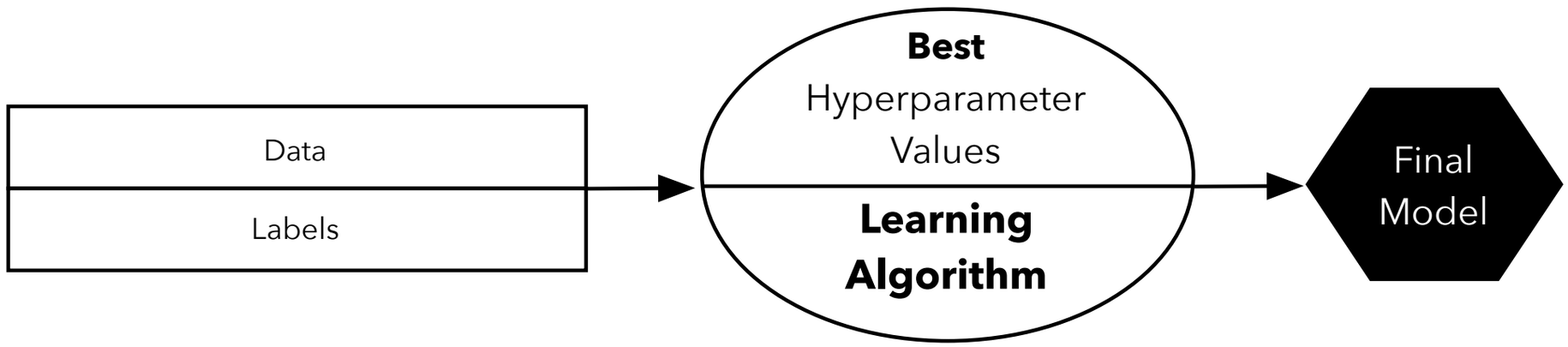
3


Training Data
Training Labels
Best Hyperparameter Values
Learning Algorithm
Model

4

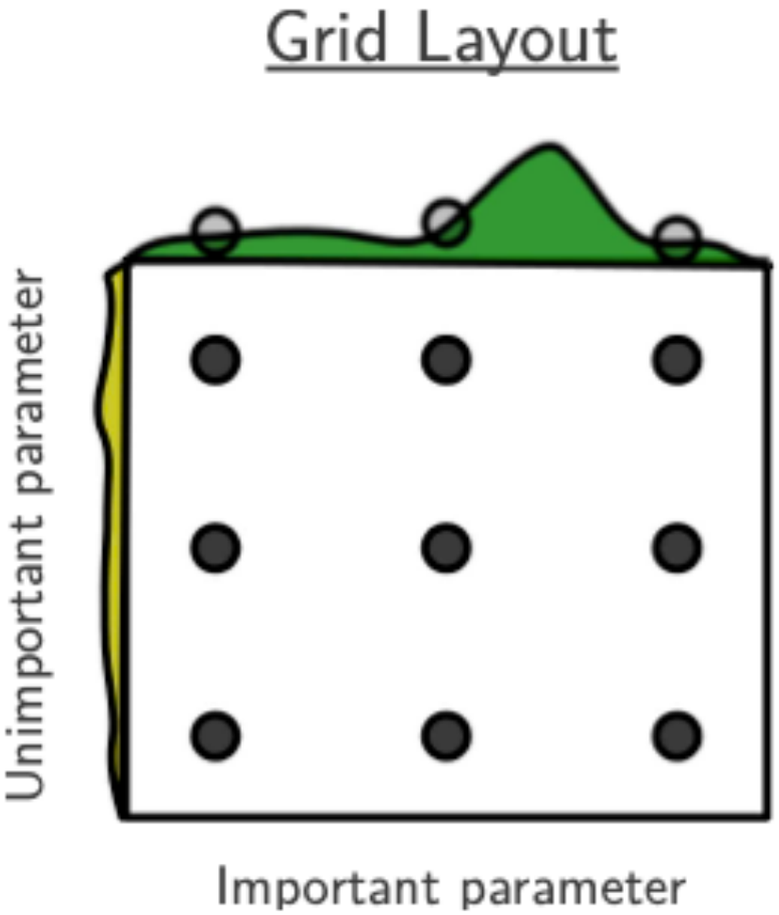

Test Data
Model
Prediction
Test Labels
Performance

5


Data
Labels
Best
Hyperparameter
Values
Learning
Algorithm
Final
Model

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

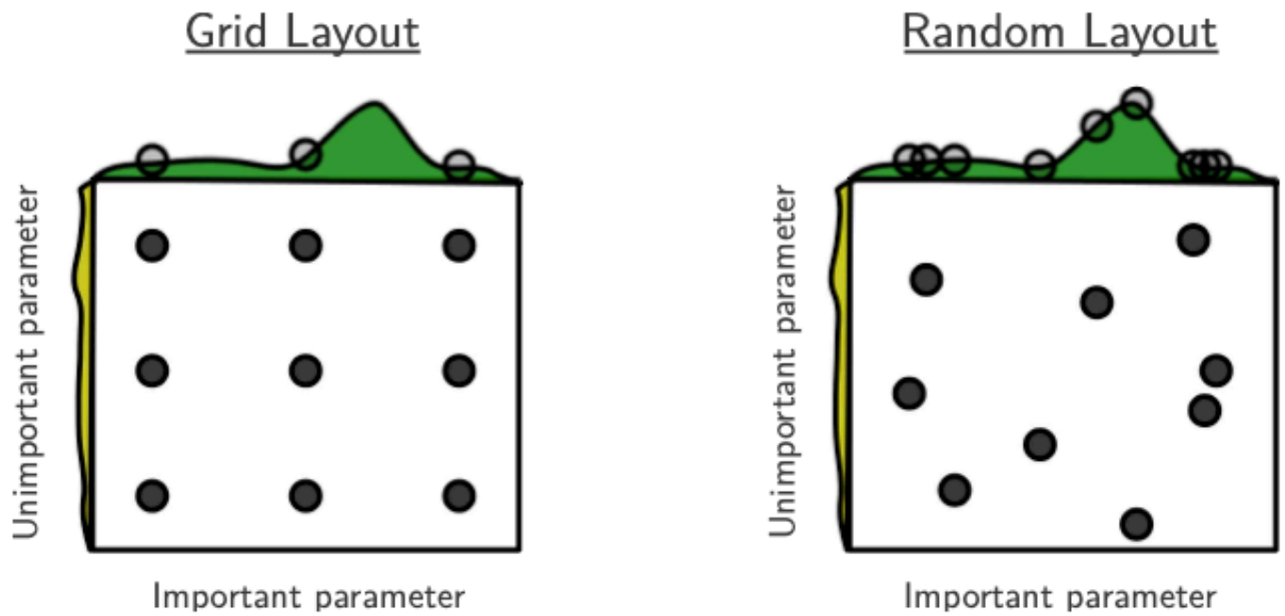


Figure 1: Grid and random search of nine trials for optimizing a function $f(x,y) = g(x) + h(y) \approx g(x)$ with low effective dimensionality. Above each square $g(x)$ is shown in green, and left of each square $h(y)$ is shown in yellow. With grid search, nine trials only test $g(x)$ in three distinct 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.

1. Lecture Overview
2. Hyperparameters
3. Cross-validation for model evaluation
4. CV for model evaluation code examples
5. Cross-validation for model selection
6. **CV for model selection code examples**
7. The 1-standard error method
8. 1std err. code examples

https://github.com/rasbt/stat451-machine-learning-fs20/blob/master/L10/code/10_06_kfold-sele.ipynb

1. Lecture Overview
2. Hyperparameters
3. Cross-validation for model evaluation
4. CV for model evaluation code examples
5. Cross-validation for model selection
6. CV for model selection code examples
7. **The 1-standard error method**
8. 1std err. code examples

The Law of Parsimony

Occam's Razor: "Among competing hypotheses, the one with the fewest assumptions should be selected."

https://en.wikipedia.org/wiki/Occam%27s_razor

The Law of Parsimony

"Simpler models are more accurate. This belief is sometimes equated with Occam's razor, but the razor only says that simpler explanations are preferable, not why. They're preferable because they're easier to understand, remember, and reason with. Sometimes the simplest hypothesis consistent with the data is less accurate for prediction than a more complicated one. Some of the most powerful learning algorithms output models that seem gratuitously elaborate -- sometimes even continuing to add to them after they've perfectly fit the data -- but that's how they beat the less powerful ones."

Pedro Domingos: "Ten Myths about Machine Learning"
https://medium.com/@pedromdd/ten-myths-about-machine-learning-d888b48334a3

The 1-standard error method

"... However, if two models perform equally well, the simpler one seems more likely (among other advantages)"

Pedro Domingos: "Ten Myths about Machine Learning"
https://medium.com/@pedromdd/ten-myths-about-machine-learning-d888b48334a3

The 1-standard error method

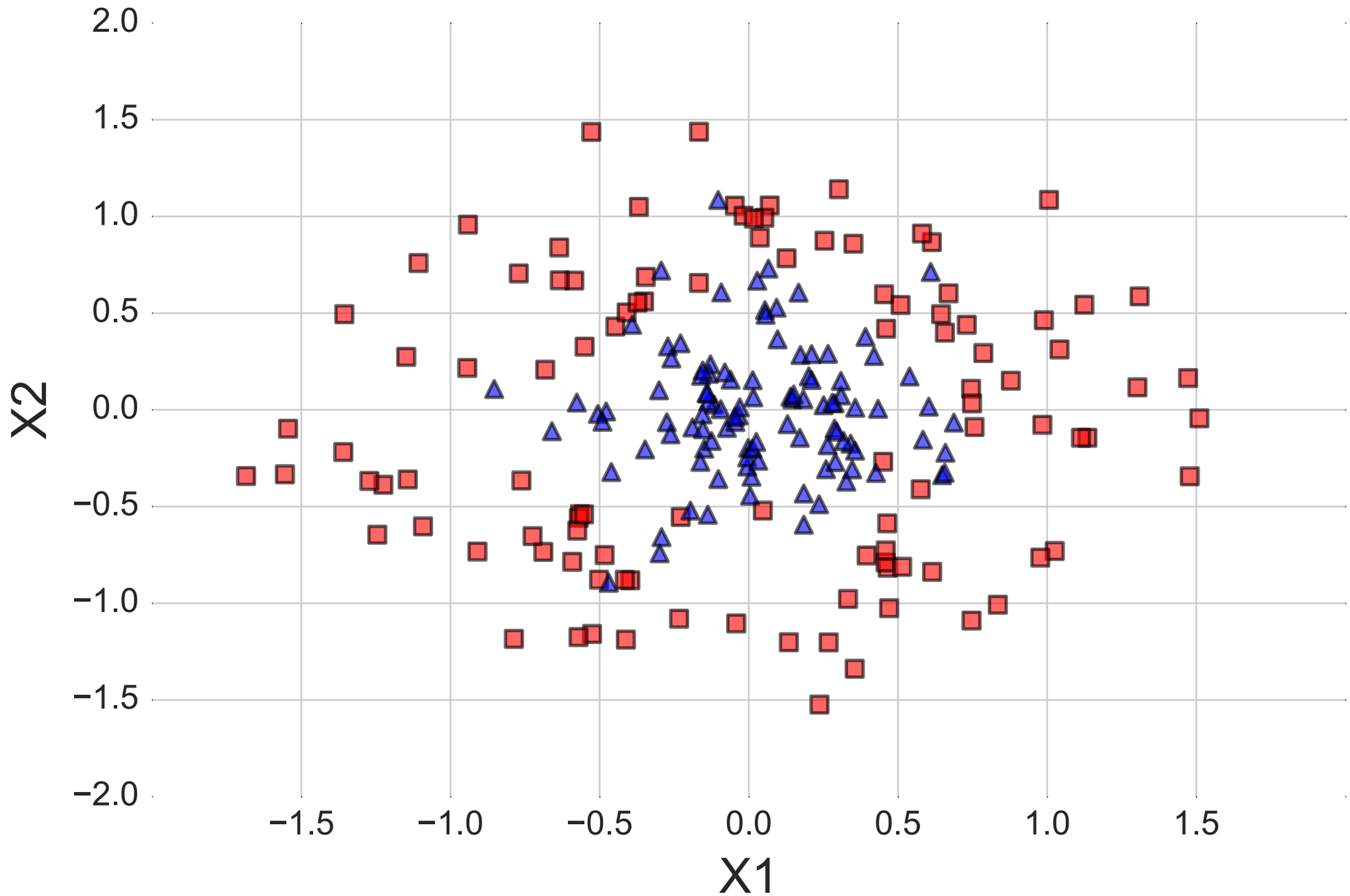
"... However, if two models perform equally well, the simpler one seems more likely (among other advantages)"

Pedro Domingos: "Ten Myths about Machine Learning"

1. Consider the numerically optimal estimate and its standard error.

2. Select the model whose performance is within one standard error of the value obtained in step 1.

The 1-standard error method



(Some toy data I generated via scikit-learn)

The 1-standard error method

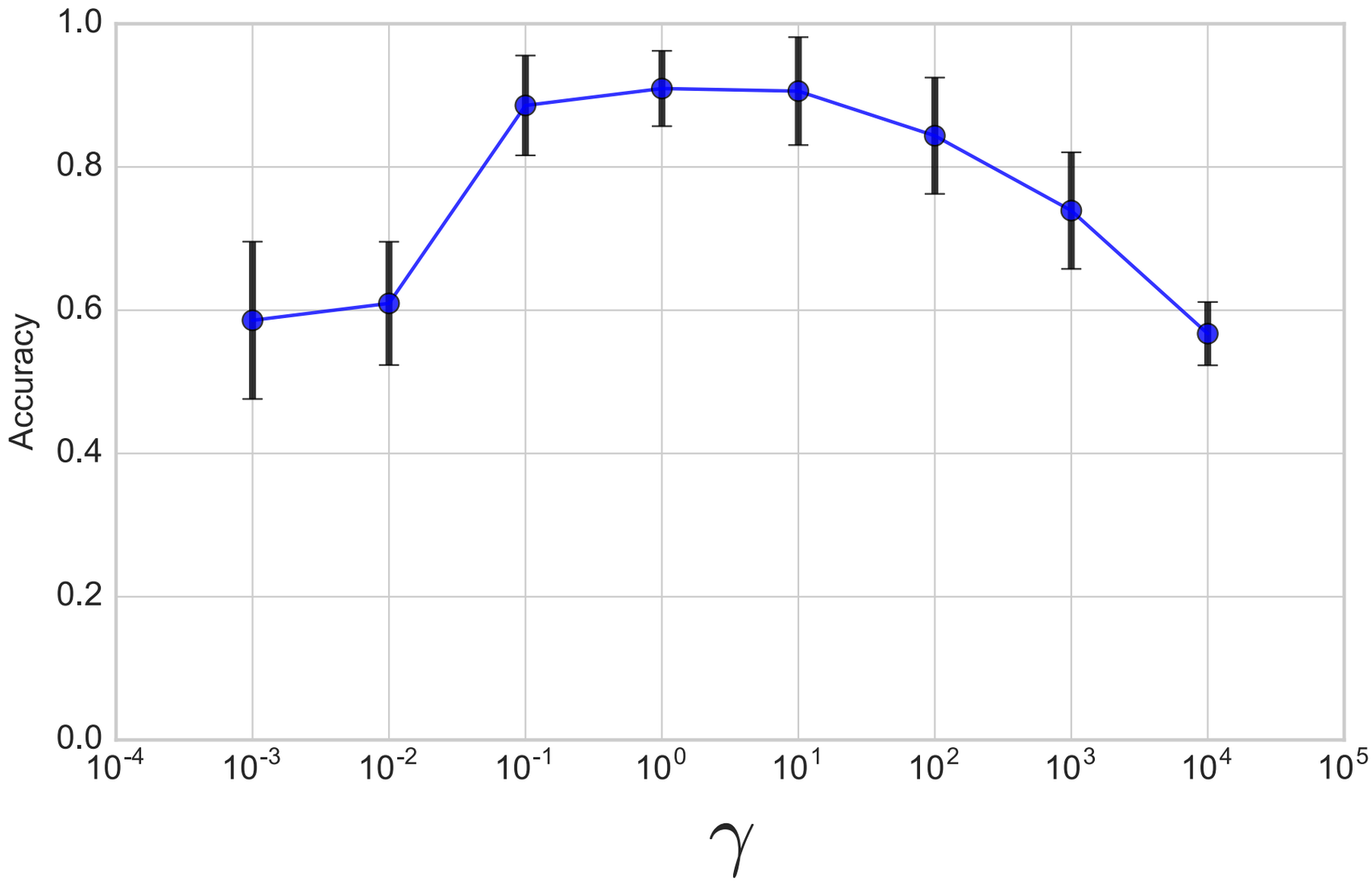
Consider a RBF-kernel SVM, where gamma controls the influence of the training points

(don't need to know the details, yet)

Gaussian/RBF-kernel: $K(x_i, x_j) = exp(-\gamma ||x_i - x_j||^2), \gamma > 0.$

The 1-standard error method

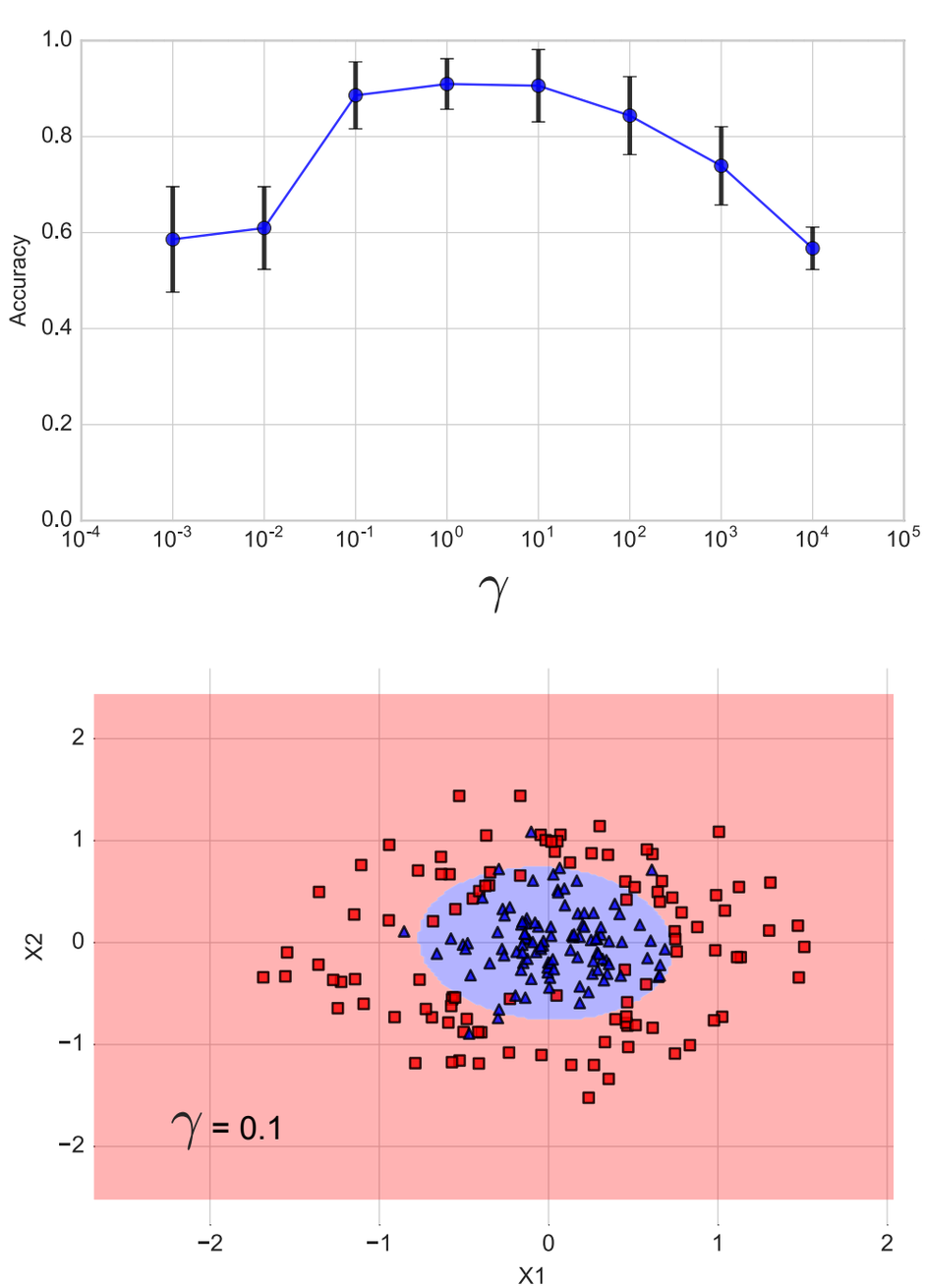
Which parameter would you select?

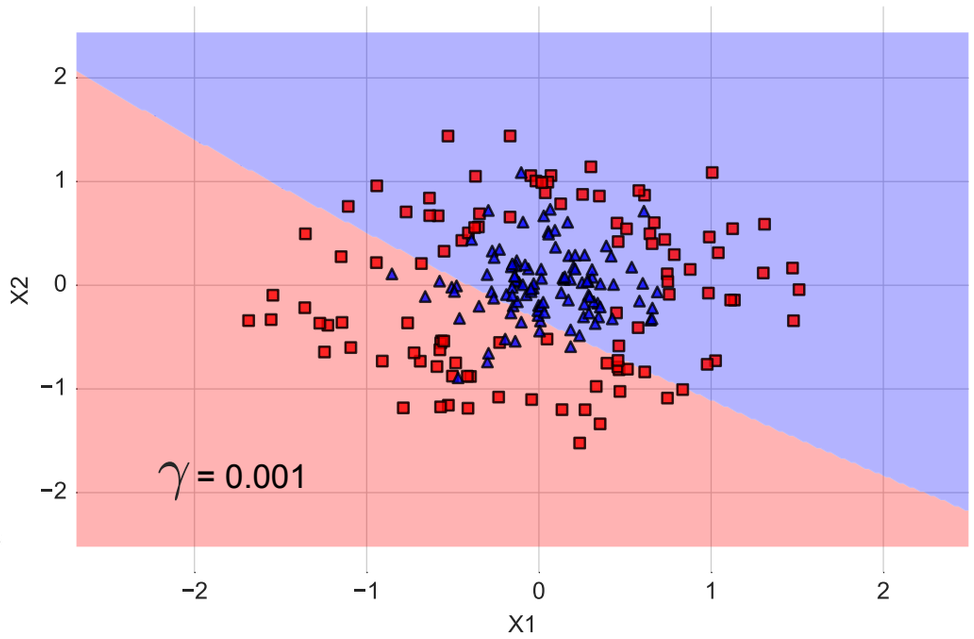


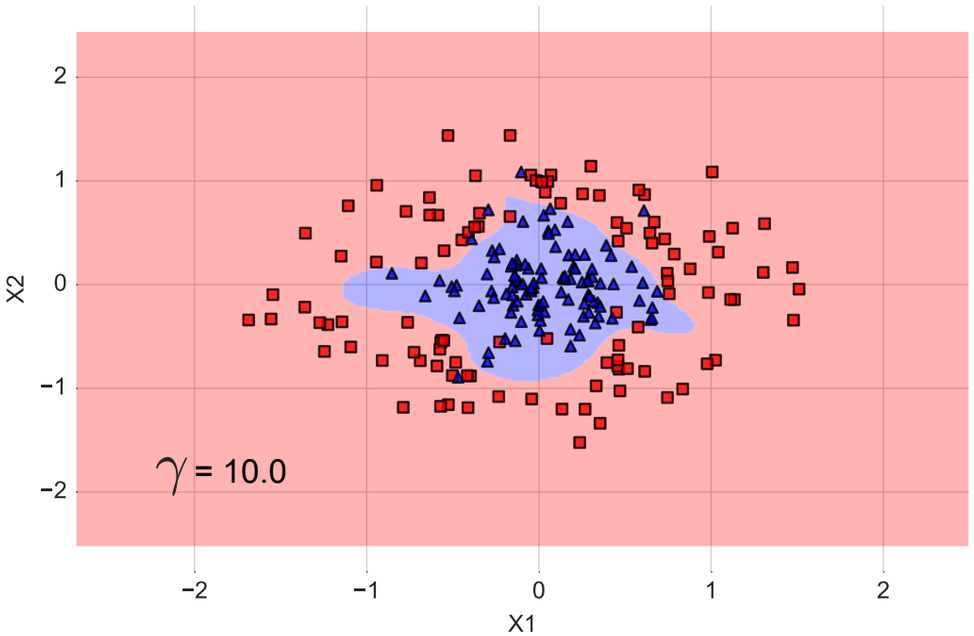
(note: here I used 10-fold CV)

The 1-standard error method

Which parameter would you select?





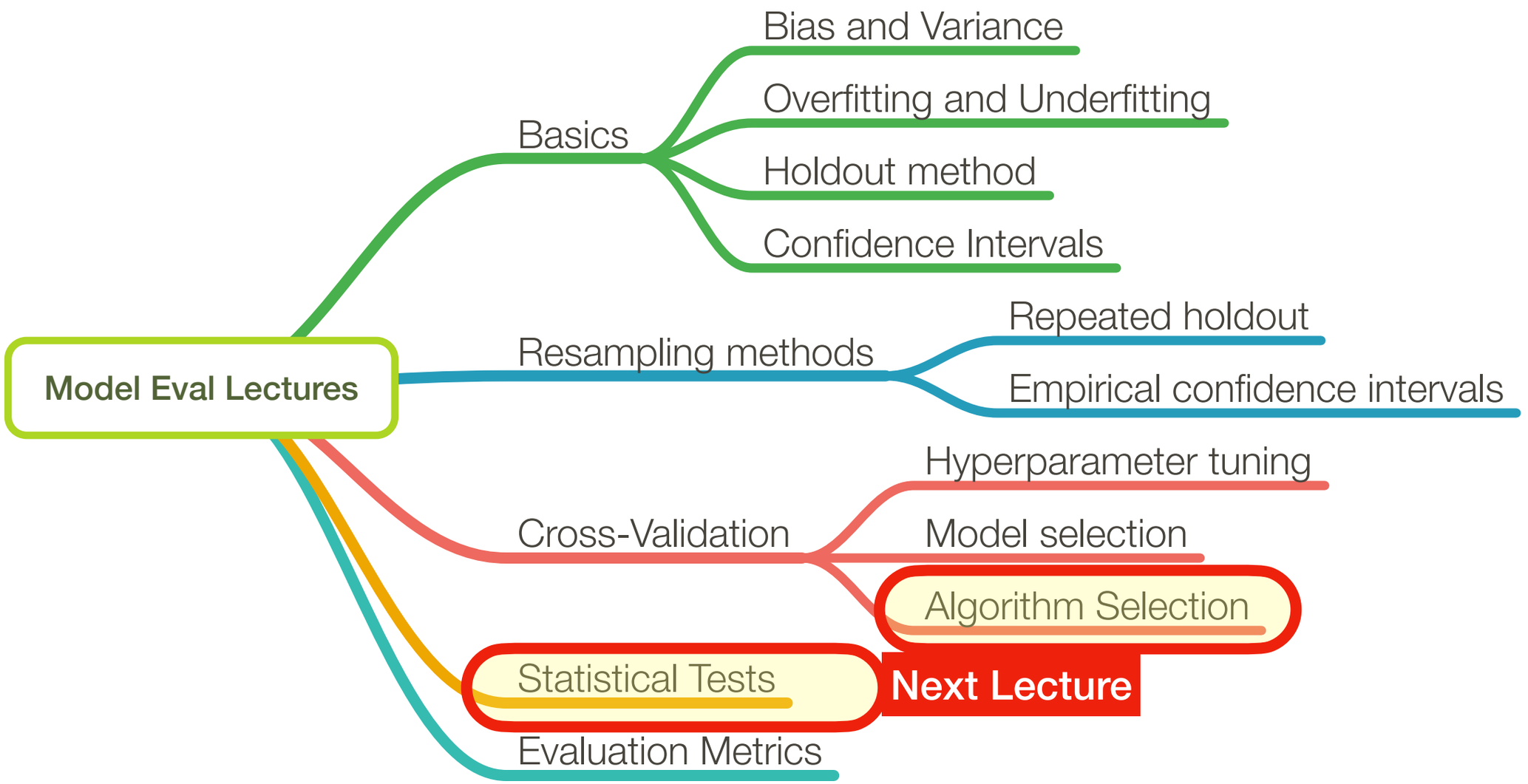


(note: here I used 10-fold CV)

1. Lecture Overview
2. Hyperparameters
3. Cross-validation for model evaluation
4. CV for model evaluation code examples
5. Cross-validation for model selection
6. CV for model selection code examples
7. The 1-standard error method
8. **1std err. code examples**

https://github.com/rasbt/stat451-machine-learning-fs20/blob/master/L10/code/10_08_1stderr.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
Next Lecture
Evaluation Metrics