Evaluating Performance II

Performance evaluation roadmap

Metrics & Evaluation
(regression/classification metrics, ROC curves)

Quantify model performance

Last Class

Experimental Design — Set of decisions to fairly compare models to determine what impacts model performance — C

Model Comparison — Fairly **compare** model generalization performance — B

Performance Evaluation — Estimate generalization performance — A

Today

Supervised Learning Performance Measurement

Regression

Classification

Binary

Multiclass

Cost / Loss Functions

- Mean squared error (MSE)
- Mean absolute error (MAE)
- Huber loss

- Cross entropy / log loss

Performance Metrics and Tools

- Root mean squared error (RMSE)
- R^2, coefficient of determination
- Mean absolute percentage error (MAPE, sMAPE)

- Classification accuracy
- True positive rate (Recall)
- False positive rate
- Precision
- F_1 Score
- Area under the ROC curve (AUC)
- Receiver Operating Characteristic (ROC) curves

- Classification accuracy
- Micro-averaged F_1 Score
- Macro-averaged F_1 Score
- Confusion matrices
- Per class metrics (recall, precision, etc.)

Performance Evaluation

Estimate model **generalization** performance by evaluating metrics on held out (meaning not in the training set) **representative** data

To ensure valid metrics of generalization performance…

1. The test data are **must be representative** of what we will encounter in practice, they cannot be a biased sample

2. The test data set **must remain separate** from our model training process in every possible way

Train-Test Split

Learning model parameters and evaluating performance

Commonly ~80/20

Used for model training: select model parameters

Evaluate generalization performance

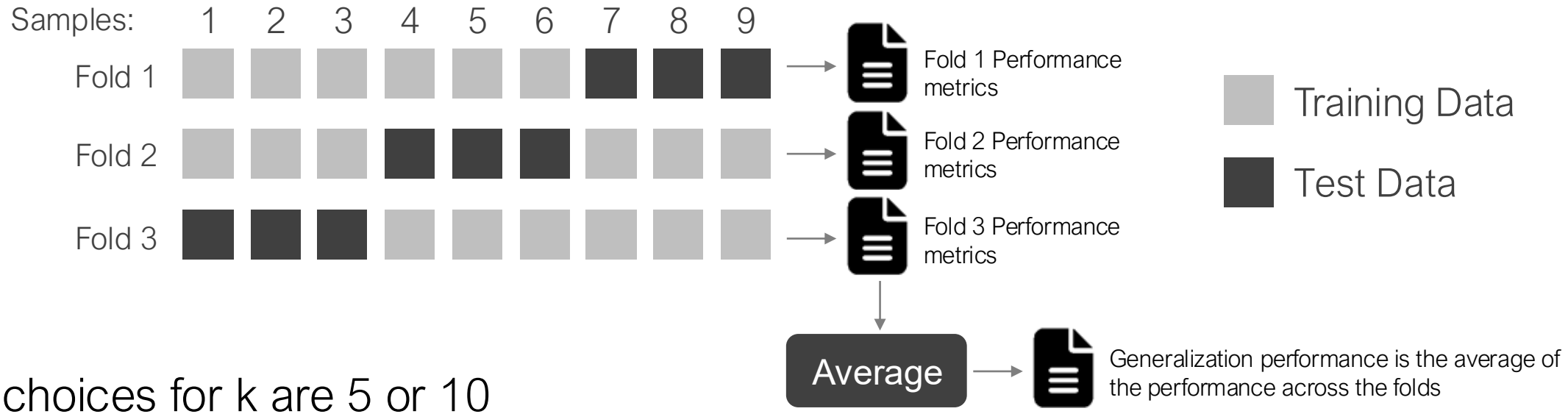
1. If our test split is too small, our estimate of generalization performance will have high variance
2. Not using all data for training produces an algorithm that is pessimistically biased
3. For small datasets, this reduction in dataset size may be detrimental

What if you have a small dataset?

K-folds cross-validation

K-fold cross validation K = 3

1 **Performance evaluation**: Train your model K times, once for each fold



Typical choices for k are 5 or 10

Average performance metrics across the splits

If k = N (number of samples): Leave-one-out cross validation

The number of splits impacts the bias-variance tradeoff of your performance estimates

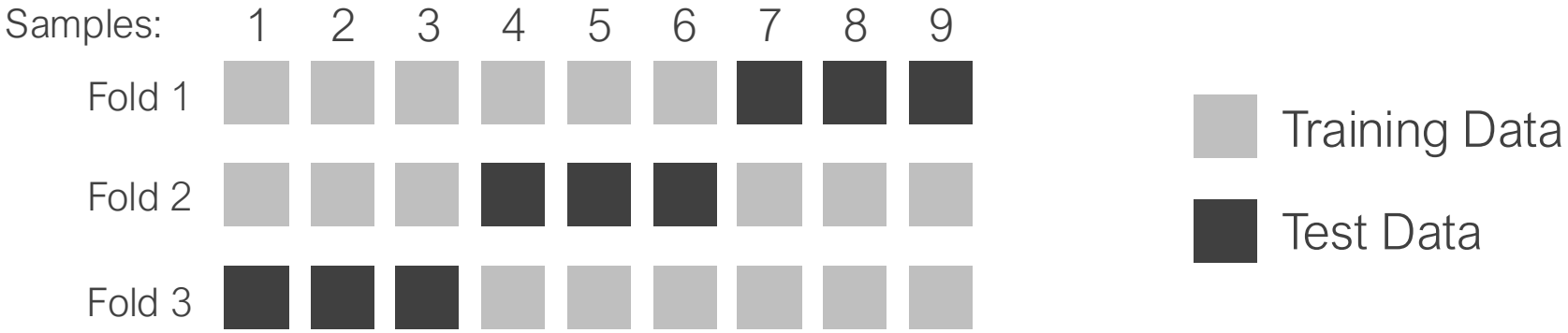
(larger k means lower bias on the performance estimate, but with higher variance)

K-folds cross validation results in **k models**

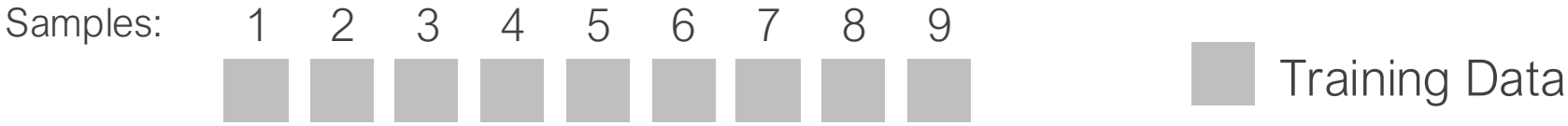
How do we apply it in practice?

After performance has been validated, train on all the data you have before you apply the model in practice

1. **Performance evaluation**: Train your model K times, once for each fold



2. **Model application**: Once you've evaluated model performance and are ready apply the model then retrain the model on ALL of your data to prepare it for unseen data



(this is not a model evaluation step, but only when you're ready to apply in practice)

Spot the misstep

1

1. Your train a logistic regression algorithm on training data

2. You evaluate the generalization performance of your trained algorithm on the training data

3. Your estimated performance is exceptional!

NEVER USE THE SAME DATA USED FOR TRAINING FOR ESTIMATING GENERALIZATION PERFORMANCE

2

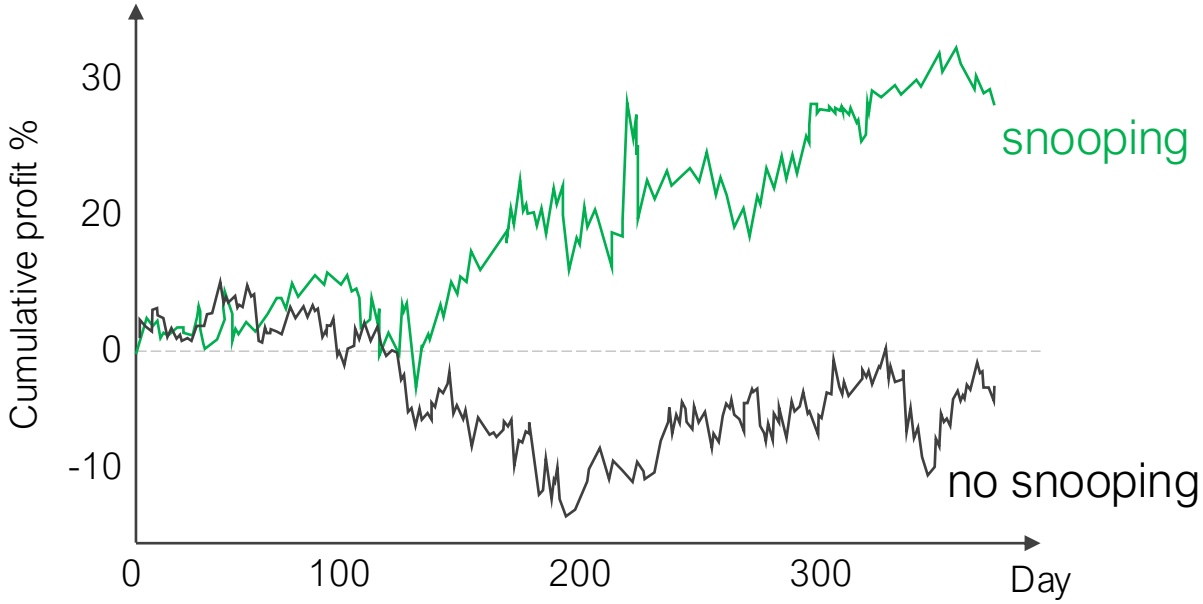
1. Goal: predict the exchange rate for the U.S. Dollar vs British Pound (using 20 past observations)

2. You take your historical data, normalize it, then split it randomly into a training and test set
   DATA SNOOPING!

3. You train on the training data, test on the test data

Results:
Your predictions are correct 56% of the time

Estimate your profits…



All preprocessing should be based on the training data alone

Abu-Mostafa, Learning From Data

3

1. Goal: predict long-term performance of a “buy and hold” strategy in stocks

2. You collect 50 years of historical data and include all companies that are currently traded in the S&P500

   SAMPLING BIAS – DATA ARE NOT REPRESENTATIVE!

3. You randomly split your data into a training and test dataset.

4. You assume you will strictly follow the “buy and hold” strategy

5. You then use apply your model on the current portfolio and predict that you will be rich in retirement!

Abu-Mostafa, Learning From Data

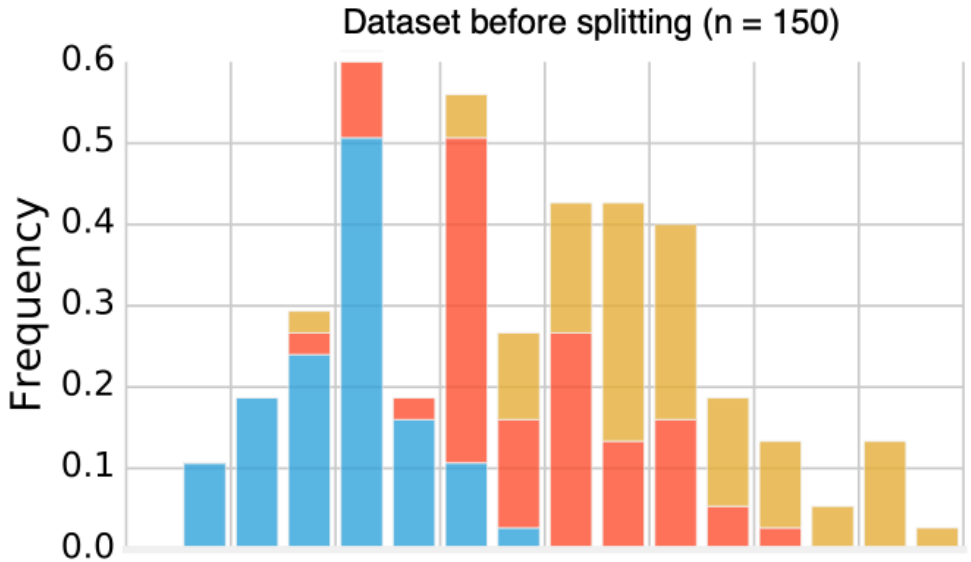
Data snooping / leakage

If a test data set has affected **any step** in the learning process, its ability to assess the generalization performance has been **compromised**.

Abu-Mostafa, Learning From Data

Sampling bias

Are the data we're using for machine learning **representative of the population you will apply on in practice**?



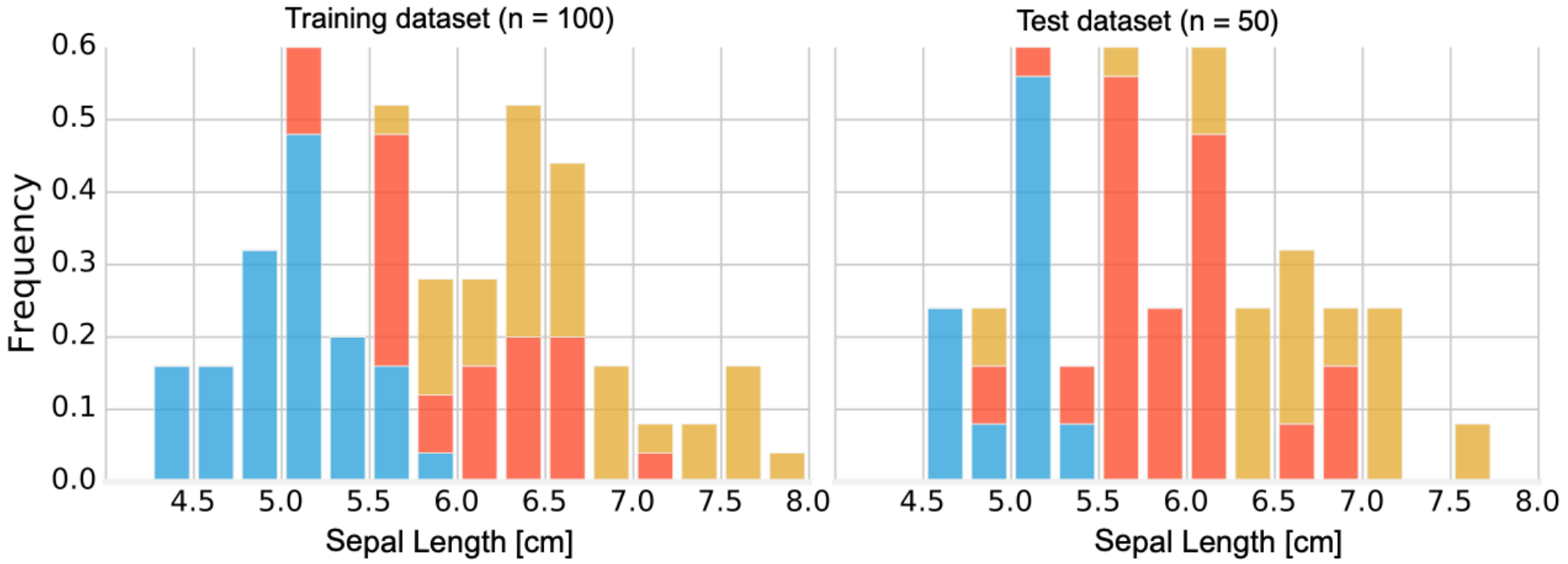
Setosa
Versicolor
Virginica

All: 50 Setosa, 50 Versicolor, 50 Virginica

Train: 38 Setosa, 28 Versicolor, 34 Virginica

Test: 12 Setosa, 22 Versicolor, 16 Virginica

One form of sampling bias



Ideally training and test sets are independent and statistically representative of the population

Dividing up your dataset we violate independence assumptions

Reduce this bias with **stratified sampling**

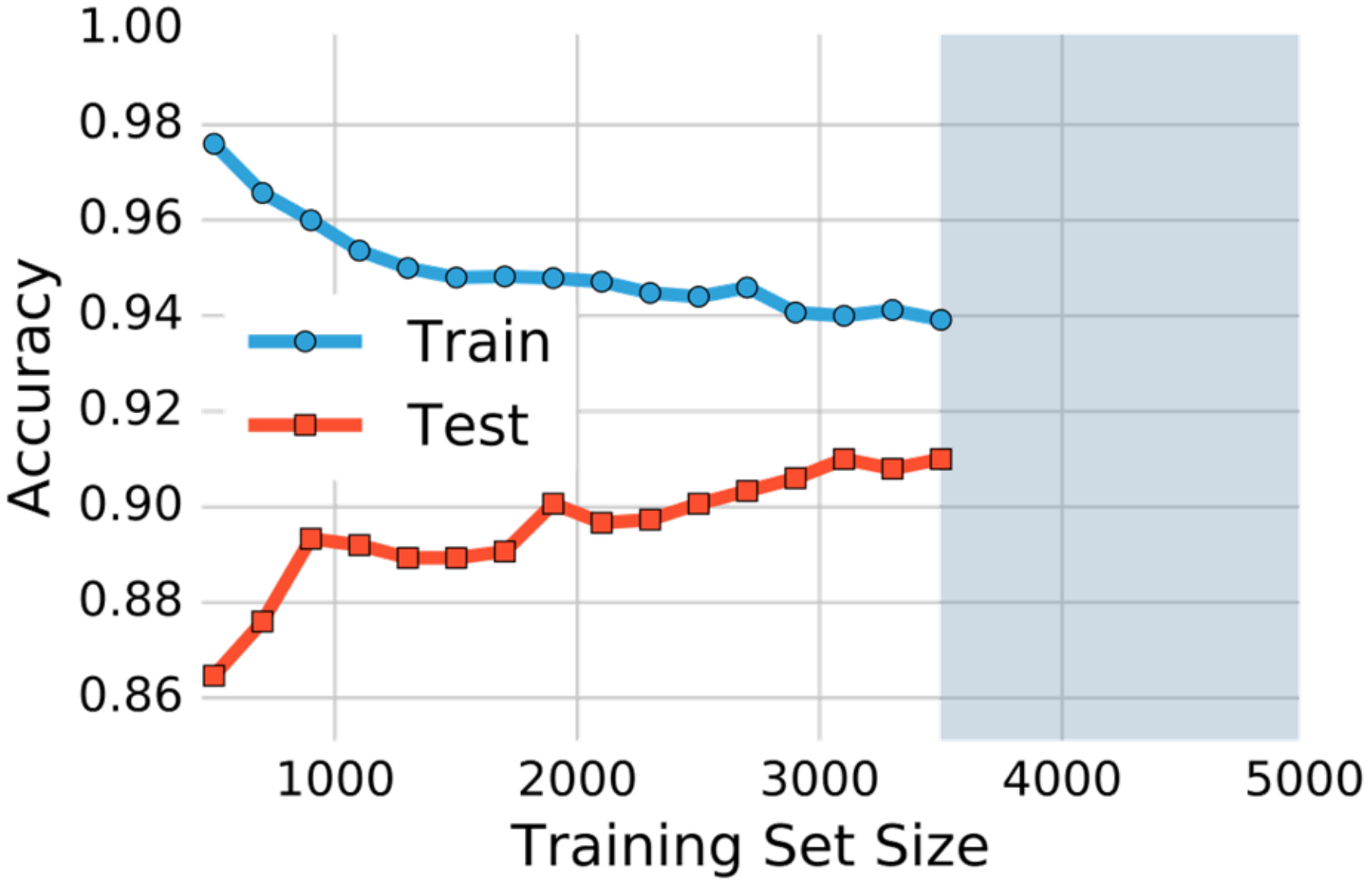
Images from Sebastian Raschka (https://sebastianraschka.com/blog/2016/model-evaluation-selection-part1.html)

Sample Size

Ideally, we would use infinite samples in our training set representing the population

In practice, we try to use as much data as possible

Larger datasets may also reduce overfit



Images from Sebastian Raschka (https://sebastianraschka.com/blog/2016/model-evaluation-selection-part2.html)

B Model Comparison

Fairly compare model performance

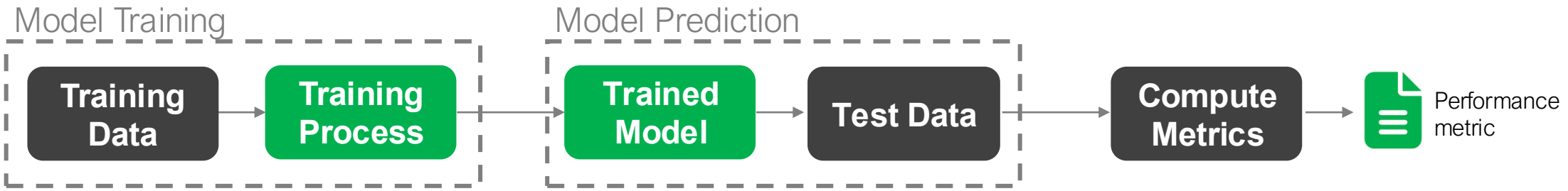
To ensure a fair model comparison…

1. The **same train/test splits** must be used for each model's performance evaluation

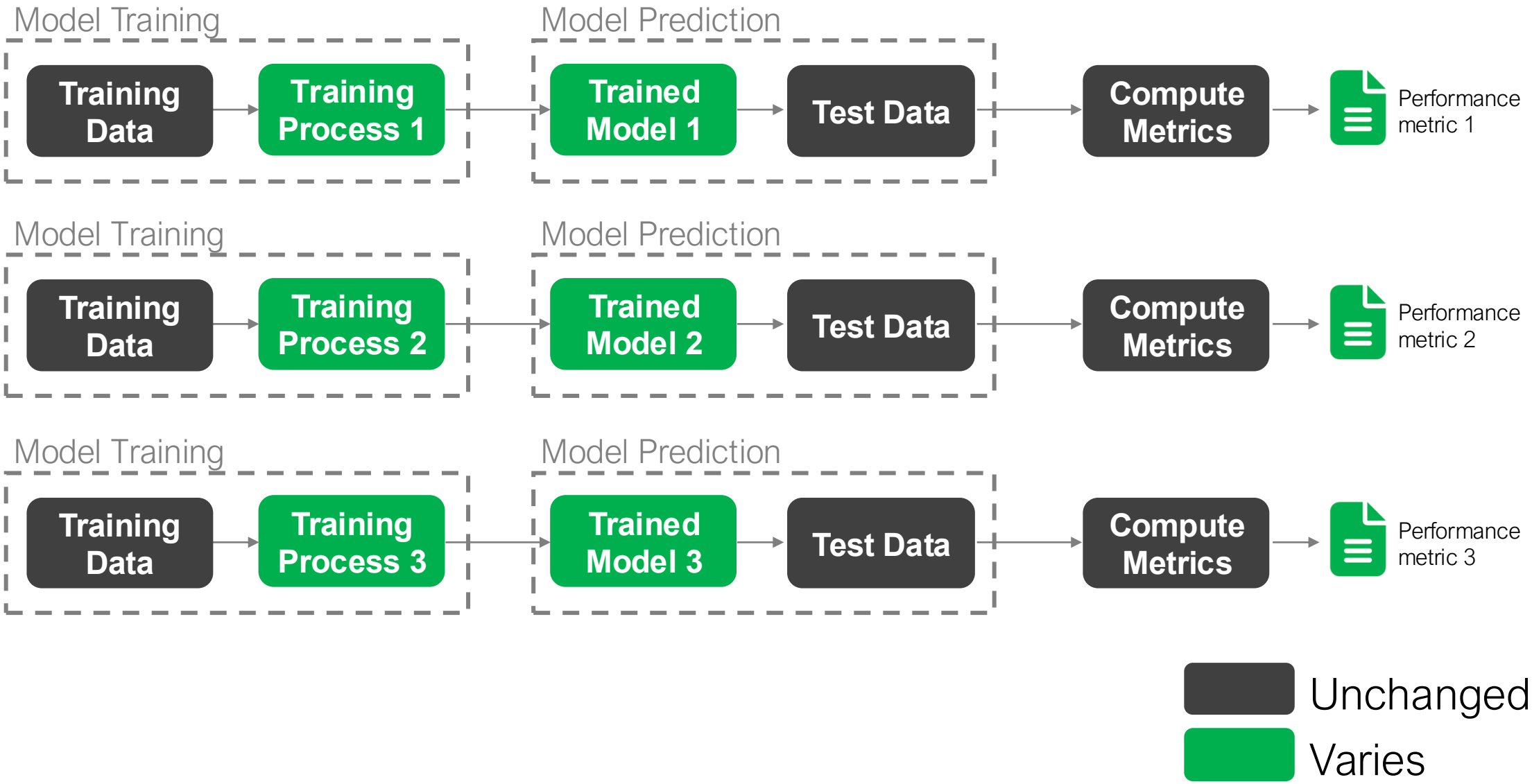
2. The **same metrics** must be used for all models

3. The models **each see the test set only once** for the final evaluation before comparison

Single model performance evaluation

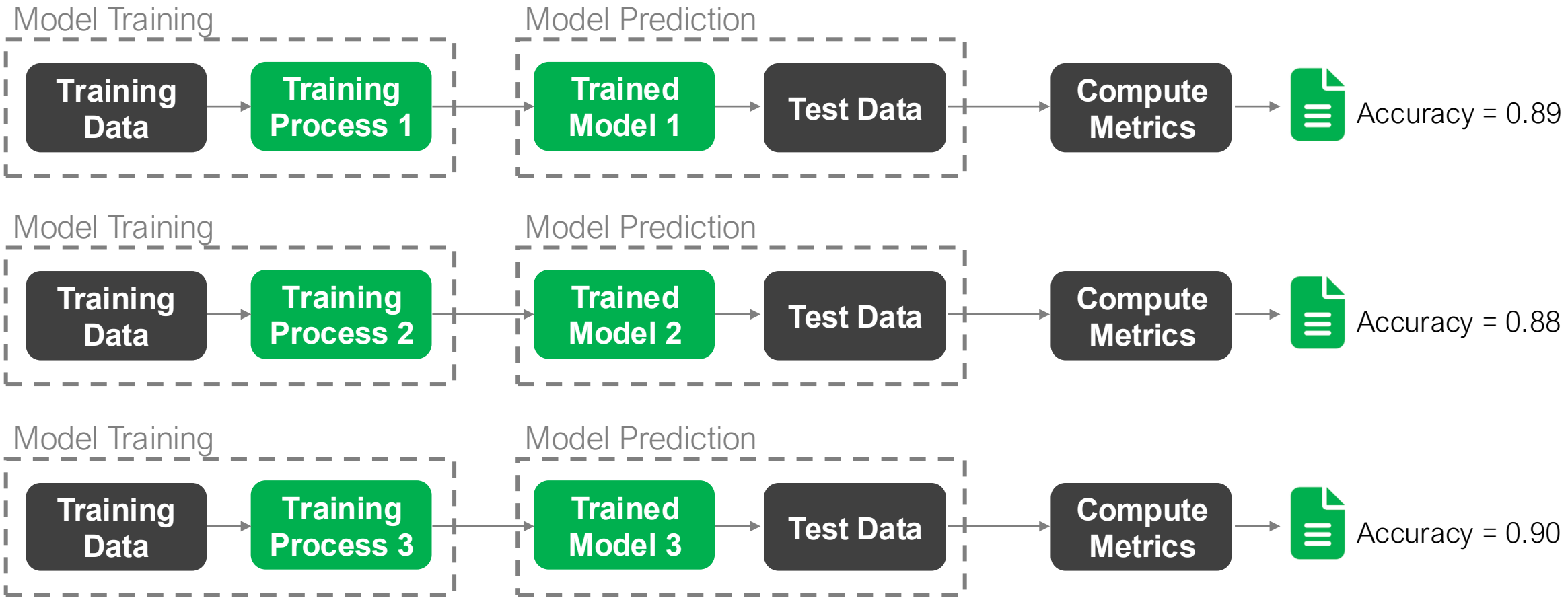


Model comparison



2 How do we use the metrics to compare models?

Model comparison: When is the performance "better"?



Is Model 3 actually better than 1 and 2?

Unchanged

Varies

We say one model is better than others when there is quantitative evidence to support that claim

Evidence is strengthened by uncertainty estimates

Performance metrics are uncertain

Data-based uncertainty

- Changes in the train/test split sampling will change metrics
- Changes in training data will change model performance
- Changes in test data will change model performance

Model-based uncertainty

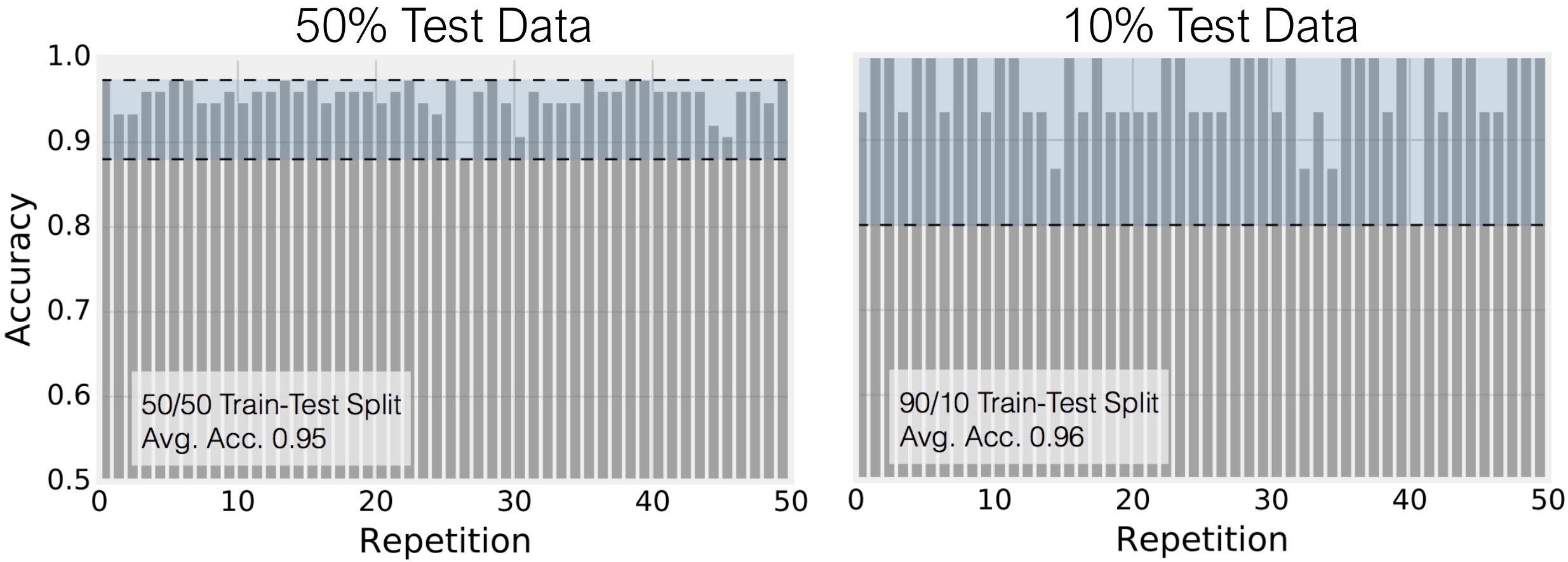
- Some models rely on stochastic model training processes (e.g., stochastic gradient descent for deep learning)
- Retraining the model on the **same training data** may result in **different predictions** on the **same test data**.

A single metric is subject to these types of uncertainty

Different data splits produce different results

Each bar represents test performance for model trained on different random splits of data

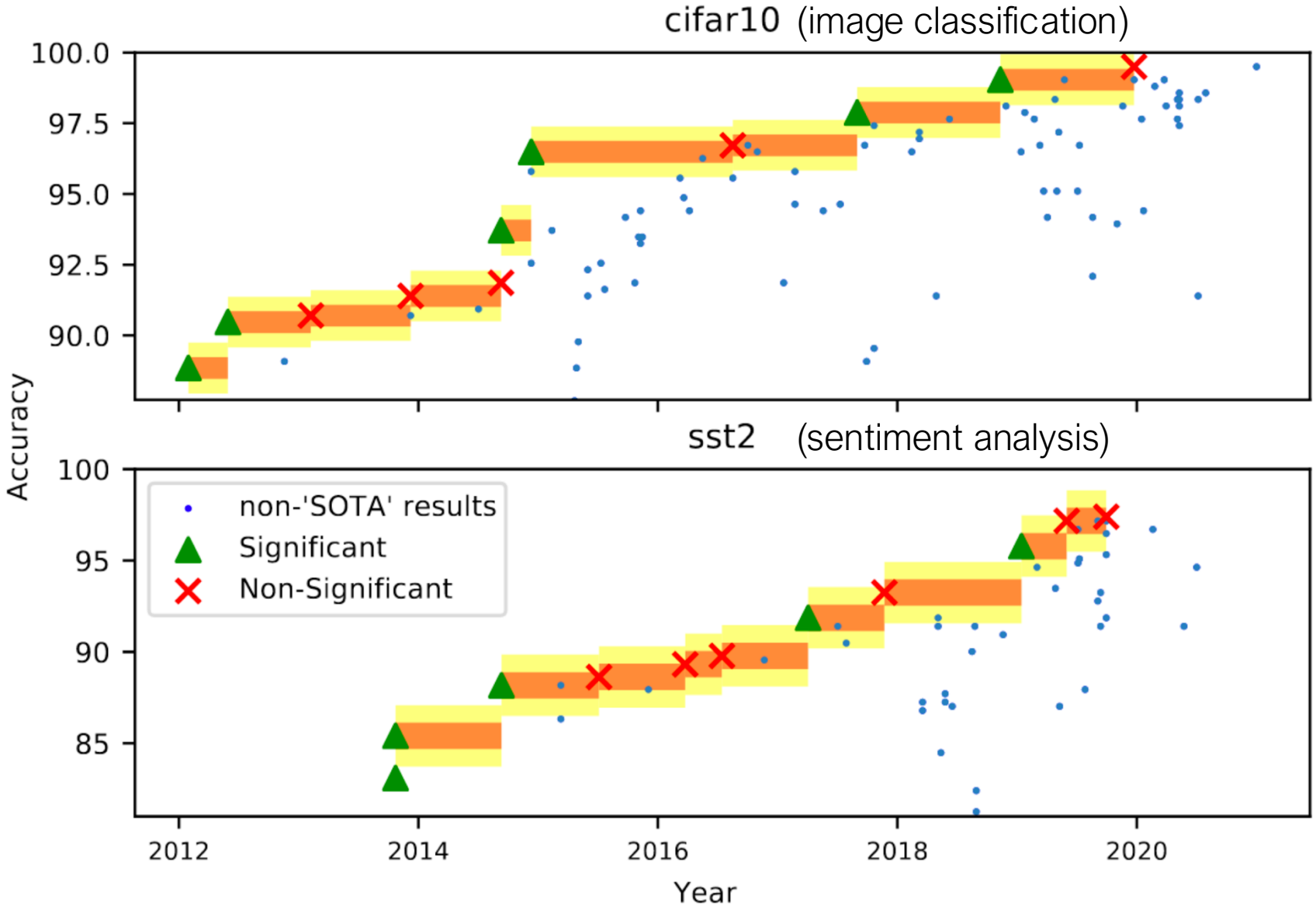
Smaller test datasets lead to greater variance in the estimate of generalization performance



Images from Sebastian Raschka (https://sebastianraschka.com/blog/2016/model-evaluation-selection-part2.html)

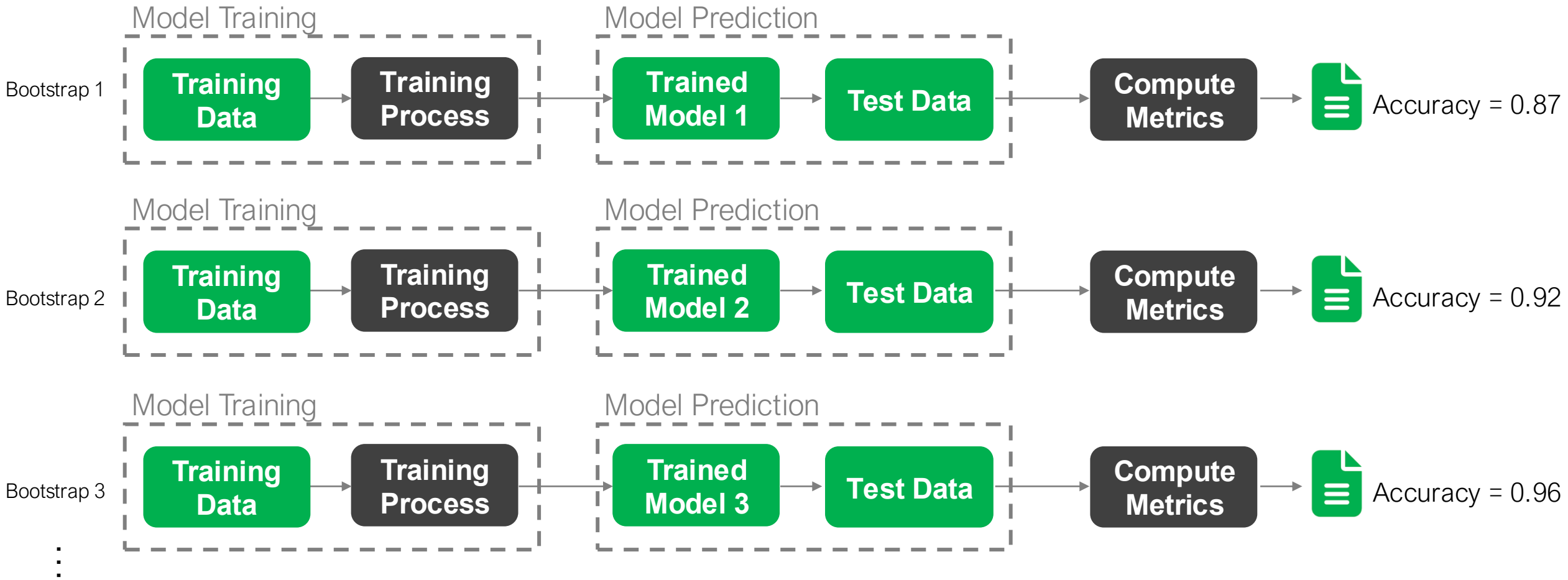
Iris dataset, KNN, k=3

Performance metrics have uncertainty



Bouthillier, X., Delaunay, P., Bronzi, M., Trofimov, A., Nichyporuk, B., Szeto, J., Sepah, N., Raff, E., Madan, K., Voleti, V., Kahou, S.E., Michalski, V., Serdyuk, D., Arbel, T., Pal, C., Varoquaux, G., Vincent, P., 2021. Accounting for Variance in Machine Learning Benchmarks. https://doi.org/10.48550/arXiv.2103.03098

Quantifying model AND data uncertainty



Repeatedly retrain the model on different train/test splits using **bootstrap sampling**

1. Training dataset is sampled with replacement
2. Test dataset is the out-of-bag data (i.e. the samples NOT selected in step 1)

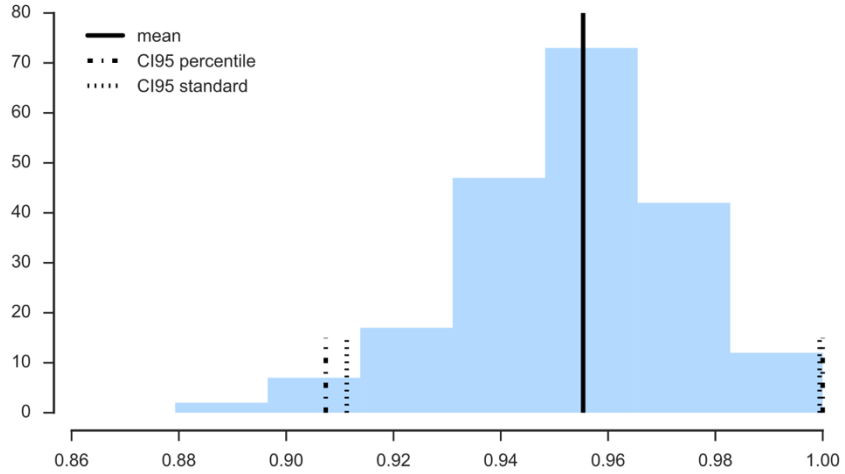
Unchanged

Varies

Quantifying model AND data uncertainty

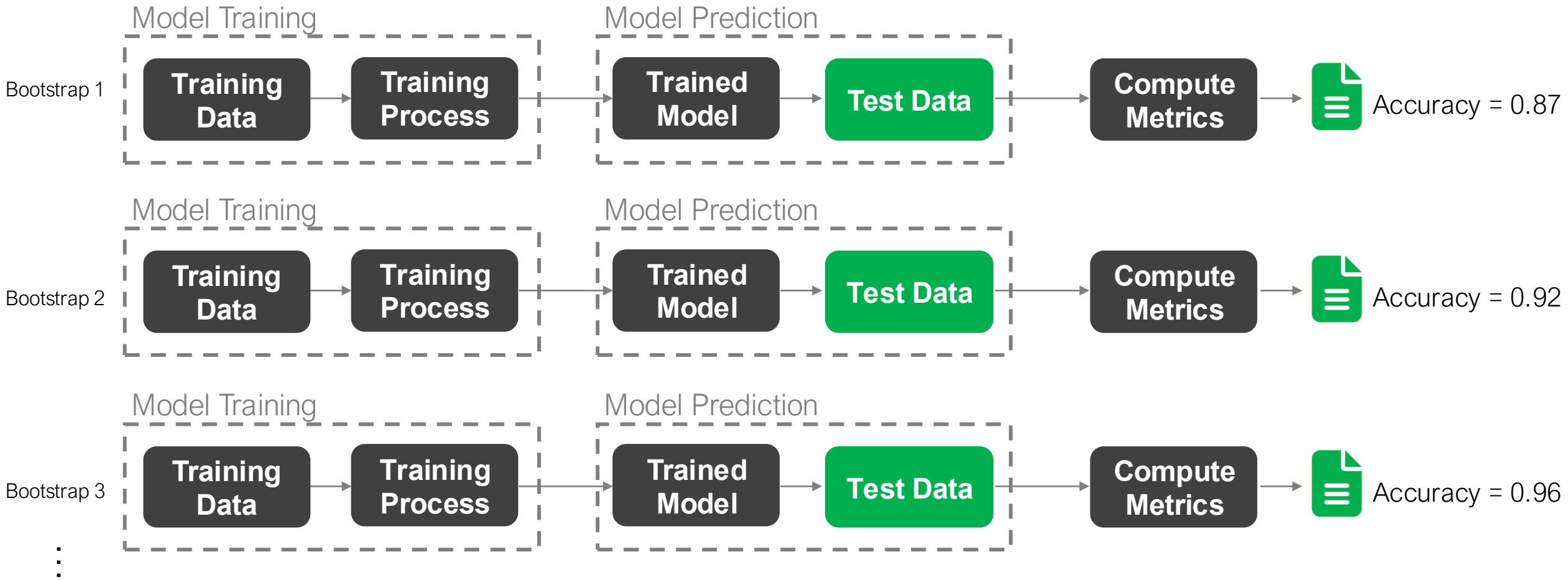
Bootstrap sampling = **with replacement**

Often used to **estimate standard errors and confidence intervals**



Images from Sebastian Raschka

Quantifying test data uncertainty

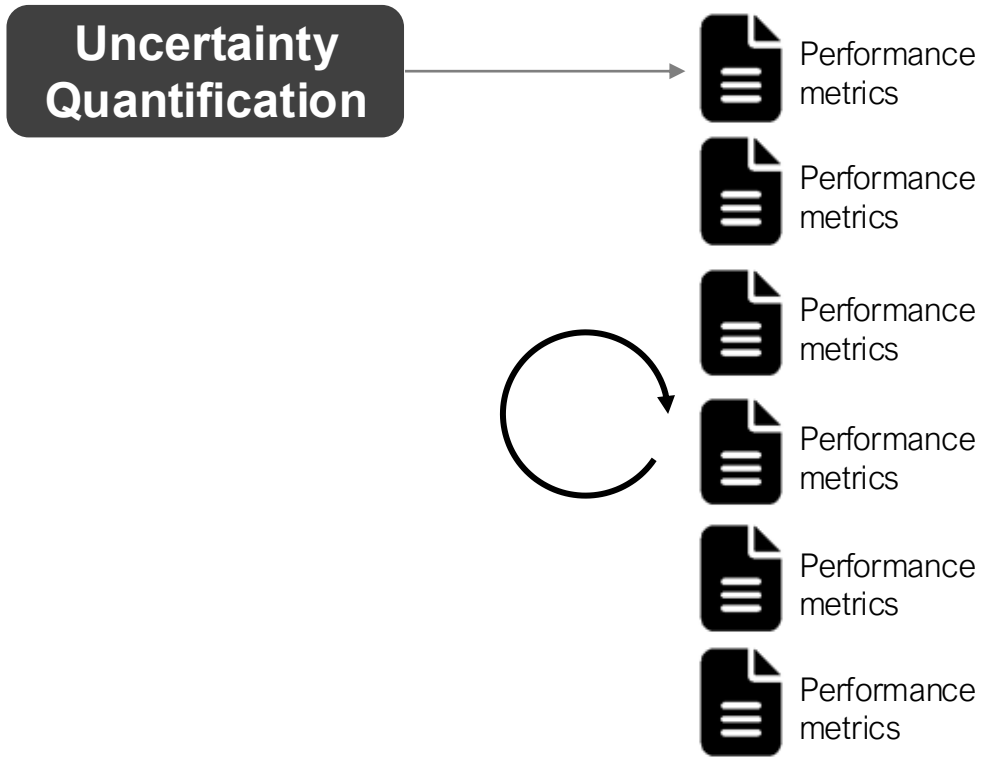


If retraining the model is cost-prohibitive, you can at least evaluate data uncertainty by bootstrap sampling the test dataset

Unchanged

Varies

Performance metric distributions



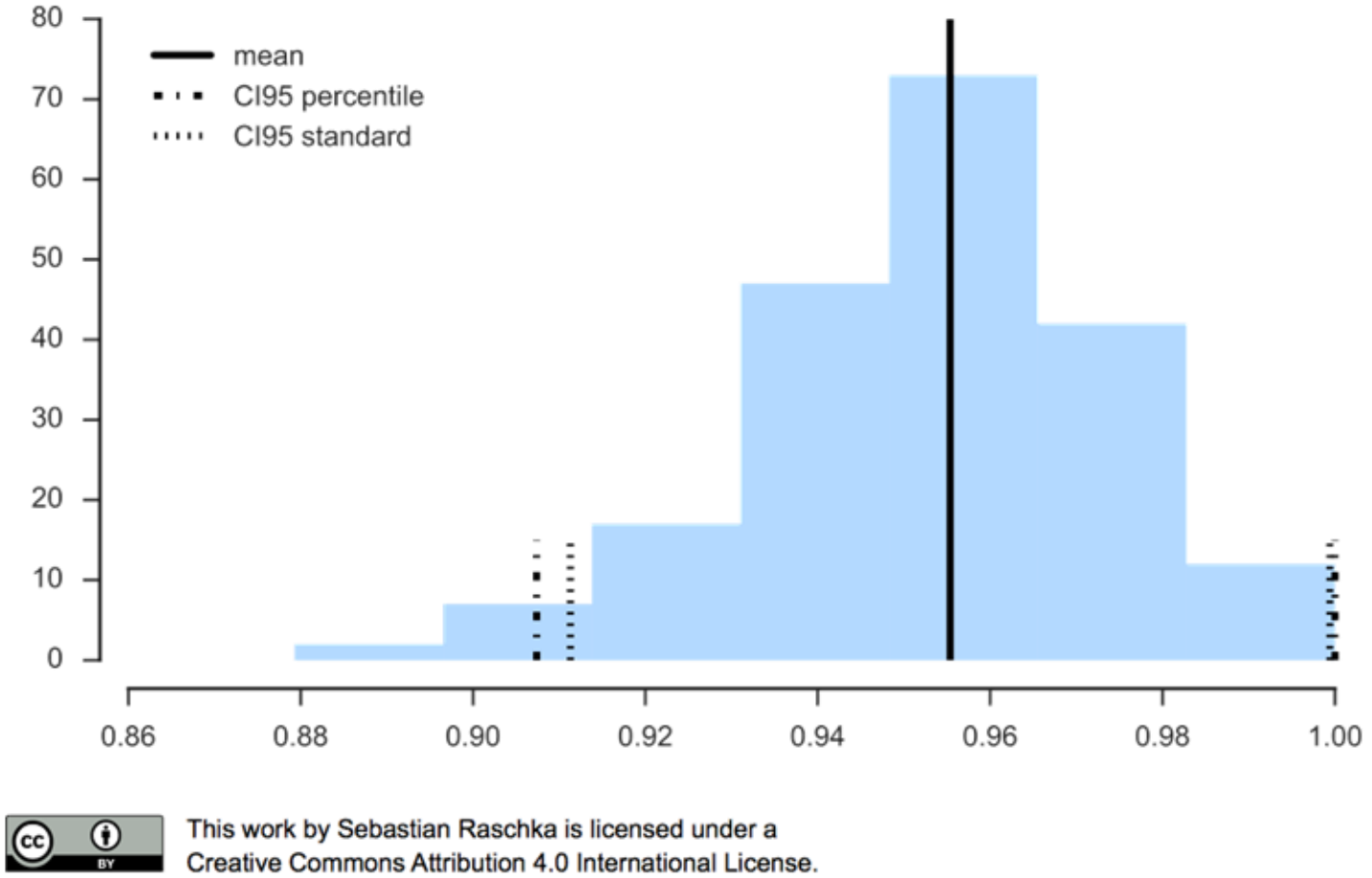


Image from Sebastia Raschka

What are Hyperparameters?

Parameters: Configuration variable that control model predictions that are adjusted during the training process based on data

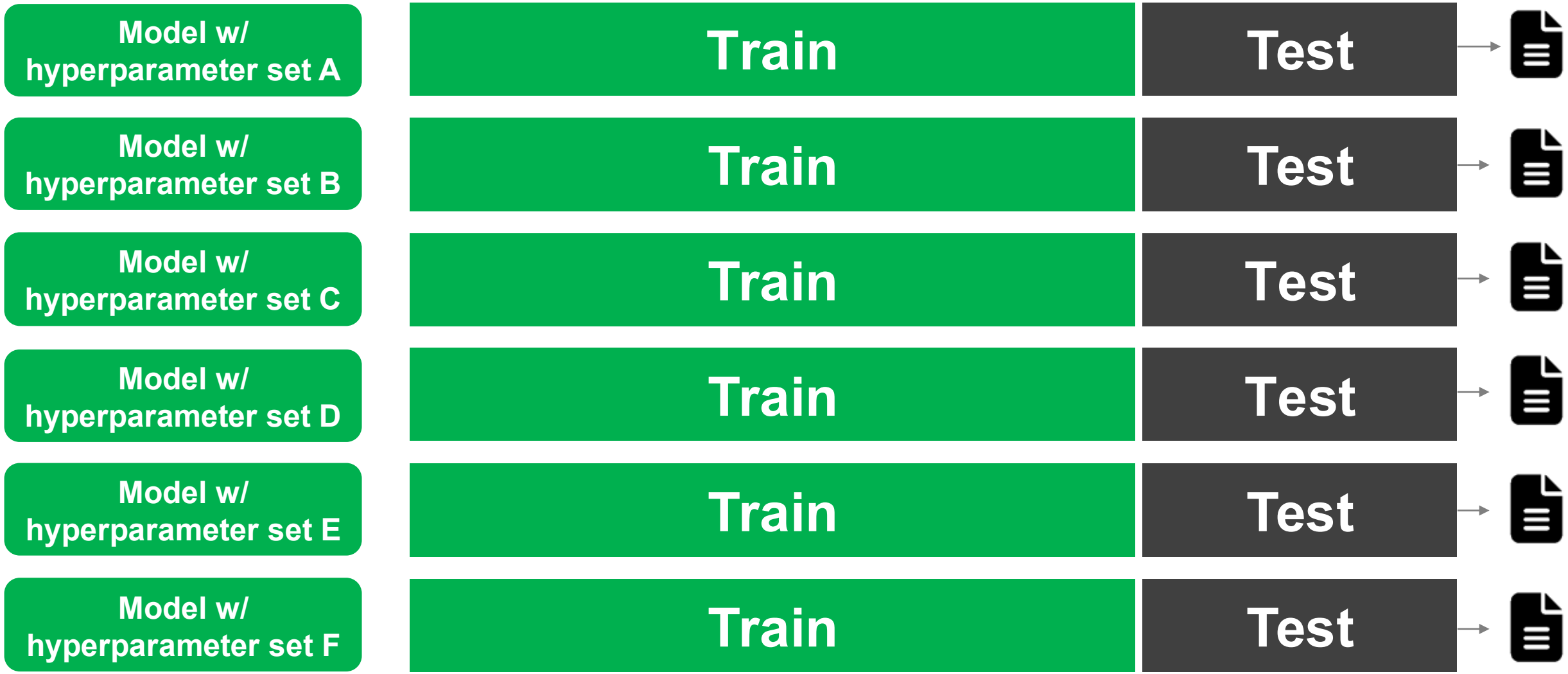
Hyperparameters: parameters set prior to model training; they are not modified during the training procedure, but often impact the training procedure.

Hyperparameter Examples

- k in KNN
- Learning rates for gradient descent of your model fitting procedure
- Model architectures (e.g. number and types of layers in neural networks)

Hyperparameter optimization

Hyperparameter tuning (comparing performance of a model across multiple values of a hyperparameter and picking the best one) **is a model comparison process**



4

1. Goal: predict the Dow Jones Industrial average

2. You randomly split your data into a training and test dataset

3. Choose a model with lots of flexibility

4. You iterate on the following process hundreds of times:
    1. Train your model on the training data
    2. Test your model on the test data
    3. Evaluate performance on the test data

    DATA SNOOPING!

5. Report that you were able to achieve 75% accuracy on your test set!

Anytime you use a held-out dataset for model comparison, it can no longer be used for estimating generalization performance

Once you pick a "best" model based on the data, it biases your data towards that specific held out sample

Solution: add an additional held-out set if an unbiased generalization performance estimate is needed

Training, Validation, Test Split

Learning model parameters AND **hyperparameters** and evaluating performance

Used for model training / fitting

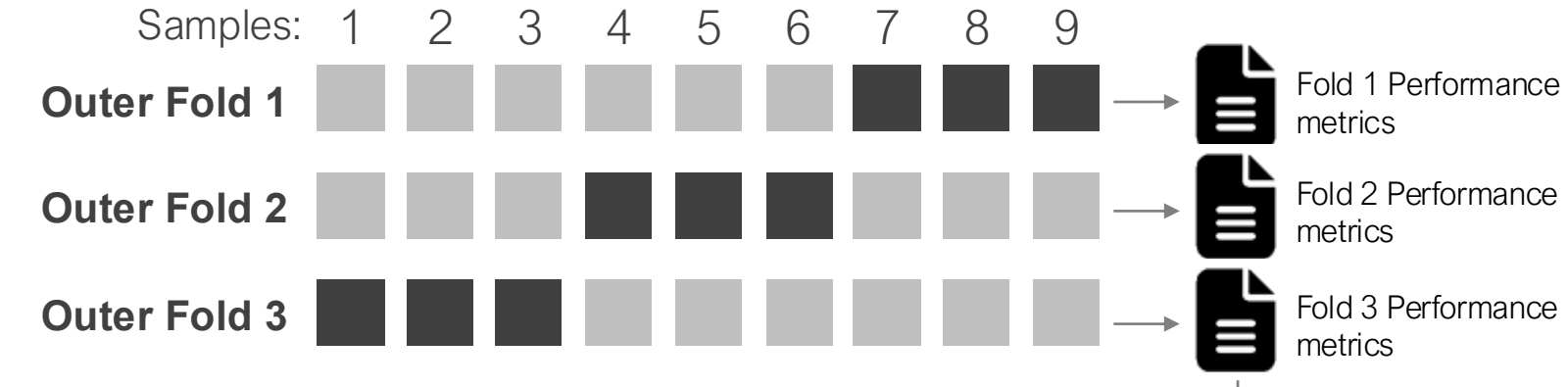
Used to compare models (including hyperparameter tuning)

Used to evaluate generalization performance of the final model(s)

Hyperparameters: parameters that control how your algorithm learns; typically set before training begins (e.g. k in KNN, learning rate, etc.)

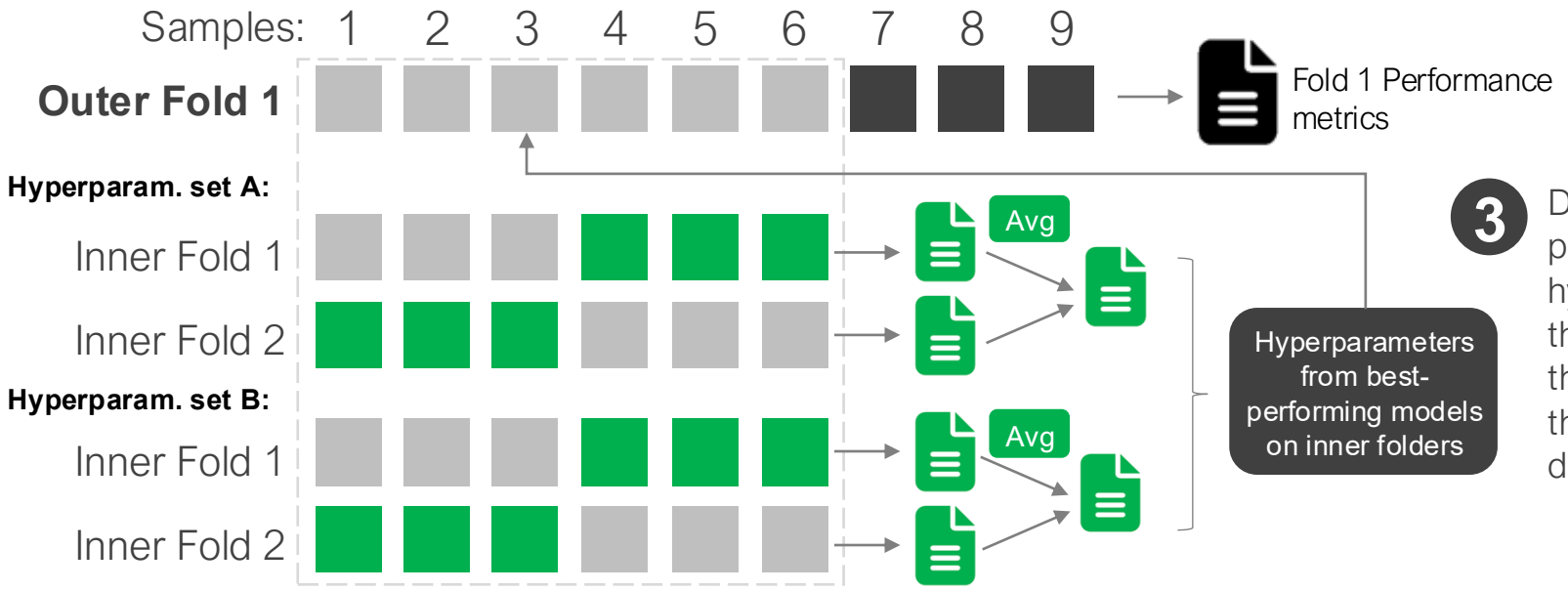
What if you need to select hyperparameters for a small dataset AND want an estimate of generalization performance?

Nested cross-validation with hyperparameters



1 For each outer fold, train your model with the best-performing hyperparameters from the inner folds

2 For the inner folds, repeatedly run K-folds, each time with a different set of hyperparameters



Inner fold = model comparison (hyperparameter tuning)
Outer fold = performance evaluation

3 Determine the best performing hyperparmeters using the inner folds and use those to train on all of the outer fold training data

Training Data
Validation Data
Test Data

When to use each technique for performance evaluation?

	no model comparison	model comparison
Large Dataset	**Train-test split**	**Train-validation-test split**
Small Dataset	**Cross-validation**	**Nested Cross-validation**

Note: hyperparameter optimization can be considered a form of model comparison

C Experimental design

Set of decisions to fairly compare models to determine what impacts model performance

Changing **ONE** aspect at a time holding **EVERYTHING ELSE** constant

Example Experimental Design

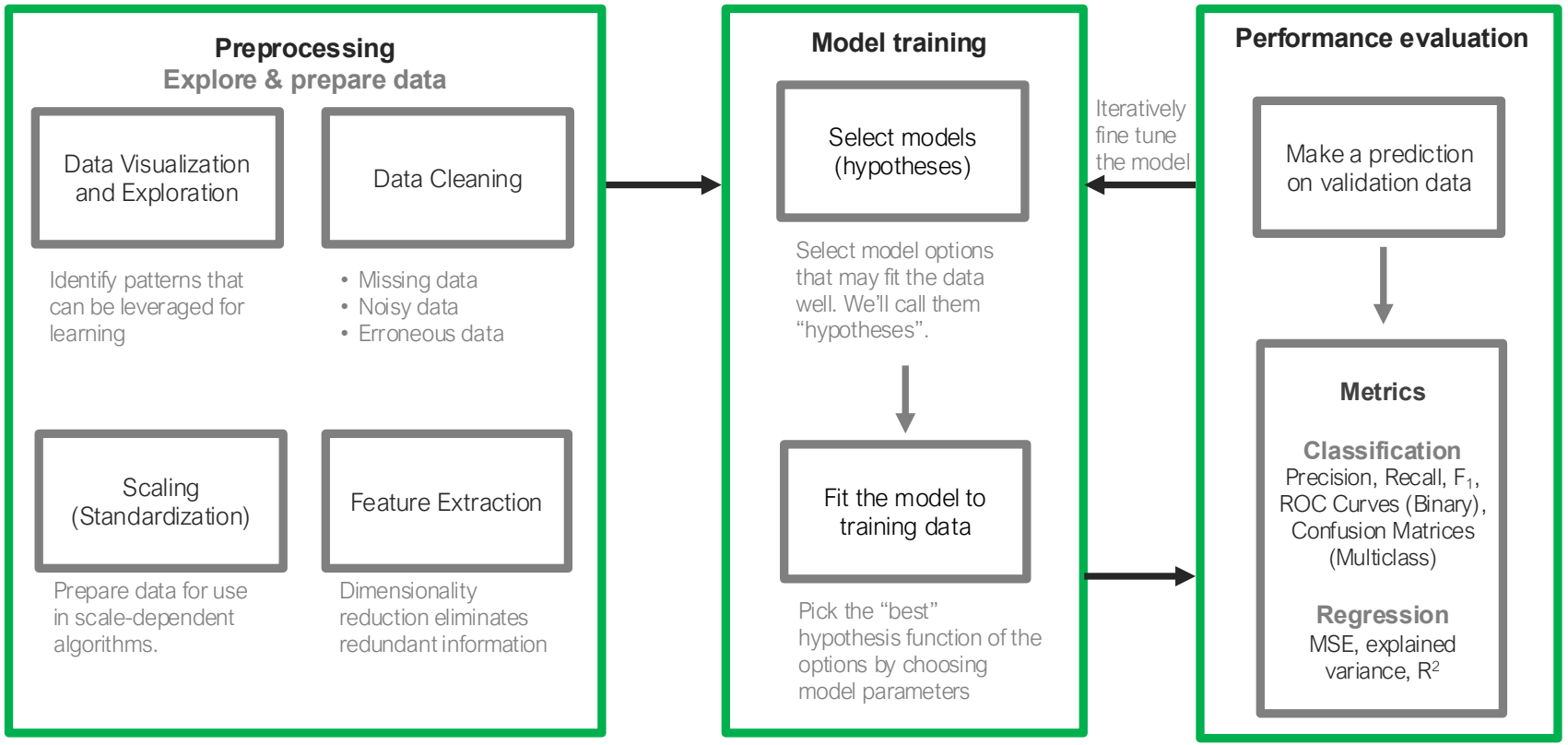
Enables us to better understand why a model behaves the way it does, typically by comparing models (e.g., does Model A or Model B perform better)

Requires that you ONLY vary one characteristic at a time (e.g. model architecture, hyperparameters, training data)
(unless you are actively looking to estimate the variation induced by that characteristic)

Requires that you control for uncertainty (in the model AND in the data), wherever possible

Experimental Design

What do we need to check to make sure that there is only one change between experimental conditions?



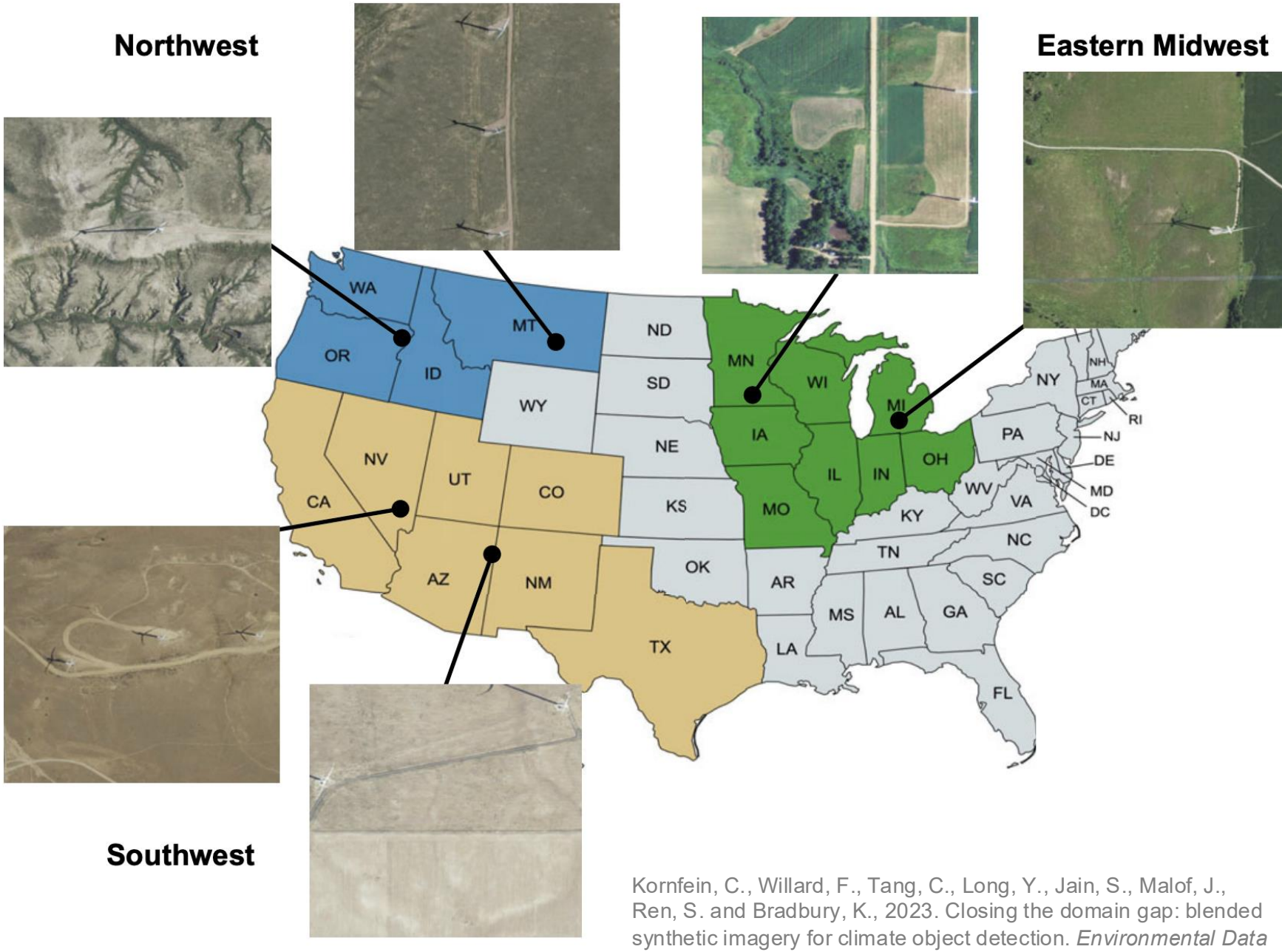
- Train/validation/test split
  - How it was split
  - The split itself
- Data imputation
- Feature selection
- Scaling
- Dimensionality reduction
- Feature transformations

- Model architecture
- Hyperparameters (e.g. learning rate, batch size, stopping criterion, choice of regularization)

Validation / test data should be the same for each model evaluation instance

Experimental Design Example

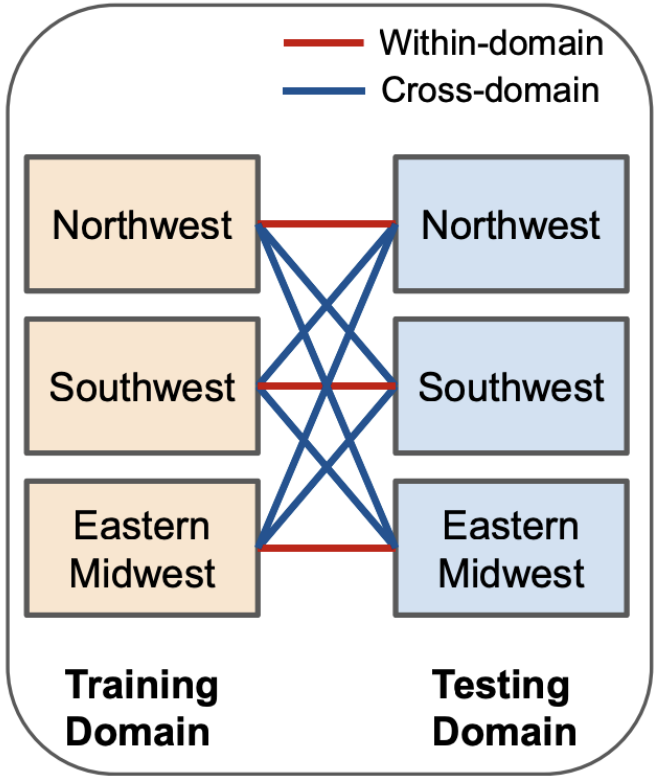
How well do models transfer to new domains?

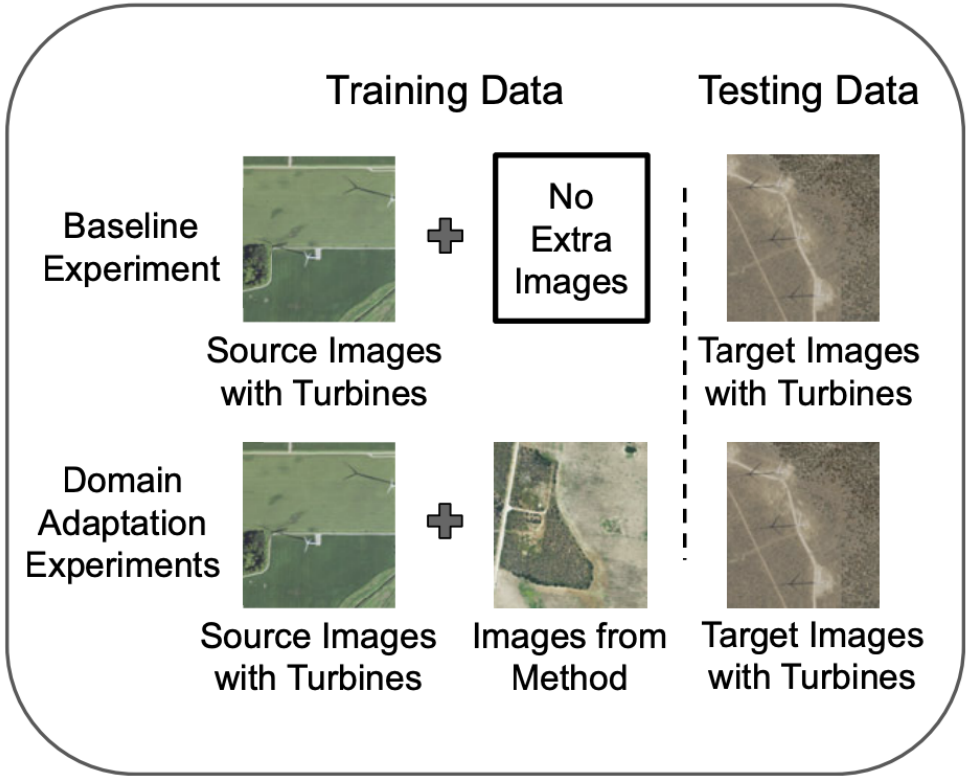


Kornfein, C., Willard, F., Tang, C., Long, Y., Jain, S., Malof, J., Ren, S. and Bradbury, K., 2023. Closing the domain gap: blended synthetic imagery for climate object detection. *Environmental Data Science*, *2*, p.e39.

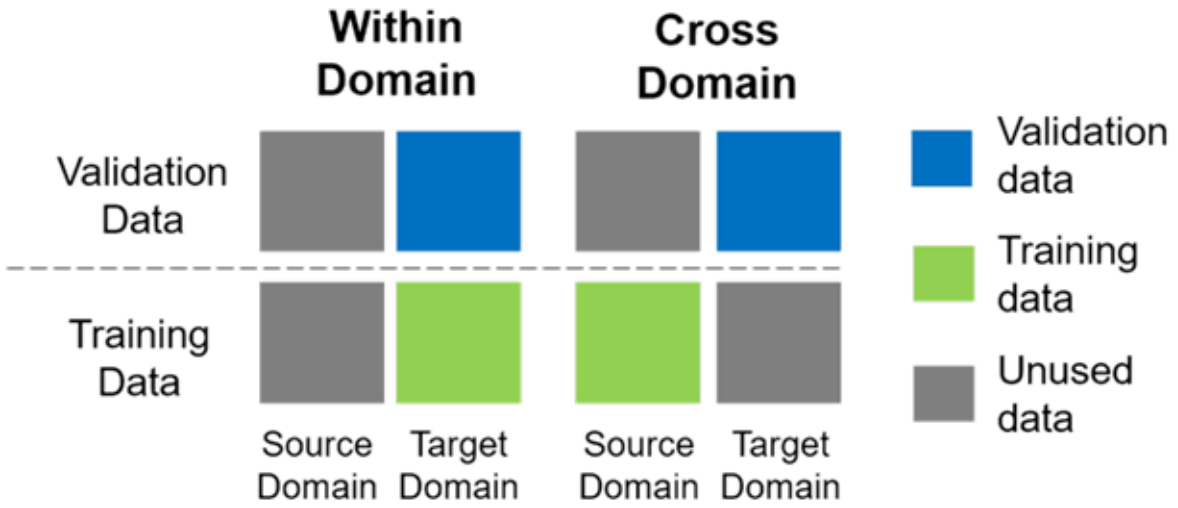
Experimental Design Example

Experimental Design
The only change between experimental conditions was the content of the training data





Training Procedure

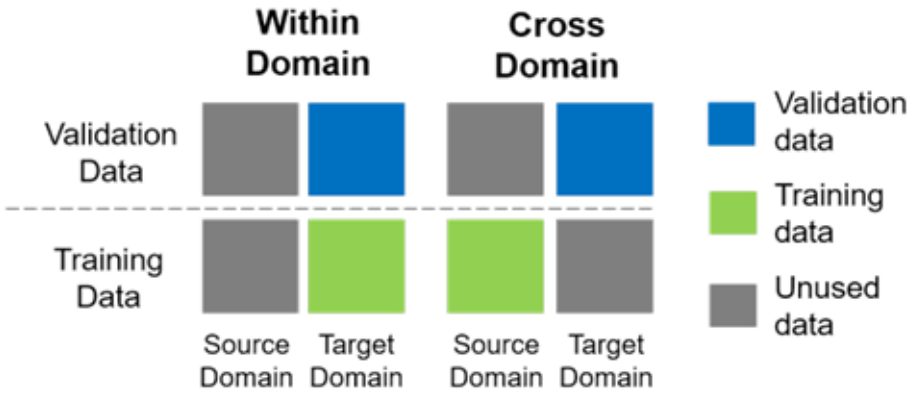


Kornfein, C., Willard, F., Tang, C., Long, Y., Jain, S., Malof, J., Ren, S. and Bradbury, K., 2023. Closing the domain gap: blended synthetic imagery for climate object detection. *Environmental Data Science*, *2*, p.e39.

Kornfein, C., Willard, F., Tang, C., Long, Y., Jain, S., Malof, J., Ren, S. and Bradbury, K., 2022. "Closing the Domain Gap–Blended Synthetic Imagery for Climate Object Detection." In 2022 Neural Information Processing Systems (NeurIPS) Climate Change AI Workshop.

Experimental Design Example

How well do models transfer to new domains?



Results shown in units of mean average precision (mAP)

Source domain	Target domain	Baseline $\pm 2\sigma$	Adding synthetic $\pm 2\sigma$	Average improvement%
EM	EM	0.822 ± 0.067	$\mathbf{0.919} \pm 0.016$	11.8%
NE		0.567 ± 0.019	$\mathbf{0.698} \pm 0.038$	23.1%
NW		0.358 ± 0.061	$\mathbf{0.424} \pm 0.114$	18.4%
SW		0.449 ± 0.160	$\mathbf{0.626} \pm 0.180$	39.4%
EM	NE	0.387 ± 0.031	$\mathbf{0.487} \pm 0.114$	25.8%
NE		0.812 ± 0.028	$\mathbf{0.842} \pm 0.013$	3.7%
NW		0.666 ± 0.061	$\mathbf{0.709} \pm 0.049$	6.5%
SW		0.412 ± 0.045	$\mathbf{0.521} \pm 0.089$	26.5%
EM	NW	0.485 ± 0.064	$\mathbf{0.521} \pm 0.054$	7.4%
NE		0.746 ± 0.018	$\mathbf{0.770} \pm 0.032$	3.2%
NW		0.895 ± 0.071	$\mathbf{0.915} \pm 0.023$	2.2%
SW		0.659 ± 0.111	$\mathbf{0.693} \pm 0.066$	5.2%
EM	SW	0.093 ± 0.016	$\mathbf{0.113} \pm 0.008$	20.9%
NE		0.121 ± 0.029	$\mathbf{0.134} \pm 0.030$	10.7%
NW		0.149 ± 0.029	$\mathbf{0.197} \pm 0.024$	32.2%
SW		0.566 ± 0.035	$\mathbf{0.568} \pm 0.104$	0.4%
Within-domain average		0.774 ± 0.050	$\mathbf{0.811} \pm 0.039$	4.8%
Cross-domain average		0.425 ± 0.054	$\mathbf{0.491} \pm 0.067$	15.7%

Bolded items represent best model (comparing a baseline model and experimental condition model)

Standard deviations contextualize model uncertainty

- This represents retraining the model multiple times to measure performance variability
- For a result to be significant, the performance change needs to be large enough to be unlikely to be due to model variability

Kornfein, C., Willard, F., Tang, C., Long, Y., Jain, S., Malof, J., Ren, S. and Bradbury, K., 2023. Closing the domain gap: blended synthetic imagery for climate object detection. *Environmental Data Science*, *2*, p.e39.

Kornfein, C., Willard, F., Tang, C., Long, Y., Jain, S., Malof, J., Ren, S. and Bradbury, K., 2022. "Closing the Domain Gap–Blended Synthetic Imagery for Climate Object Detection." In 2022 Neural Information Processing Systems (NeurIPS) Climate Change AI Workshop.

Performance evaluation roadmap

Metrics & Evaluation

(regression/classification metrics, ROC curves)

Quantify model performance

Last Class

Experimental Design

Set of decisions to fairly compare models to determine what impacts model performance

Model Comparison

Fairly **compare** model generalization performance

Performance Evaluation

Estimate generalization performance

Today

For further reading…

Raschka, Sebastian. “Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning.” *ArXiv:1811.12808 [Cs, Stat]*, November 10, 2020. http://arxiv.org/abs/1811.12808.

Kohavi, Ron. “A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection.” In *IJCAI*, 14:1137–45. Montreal, Canada, 1995. (link)