



METIS

---

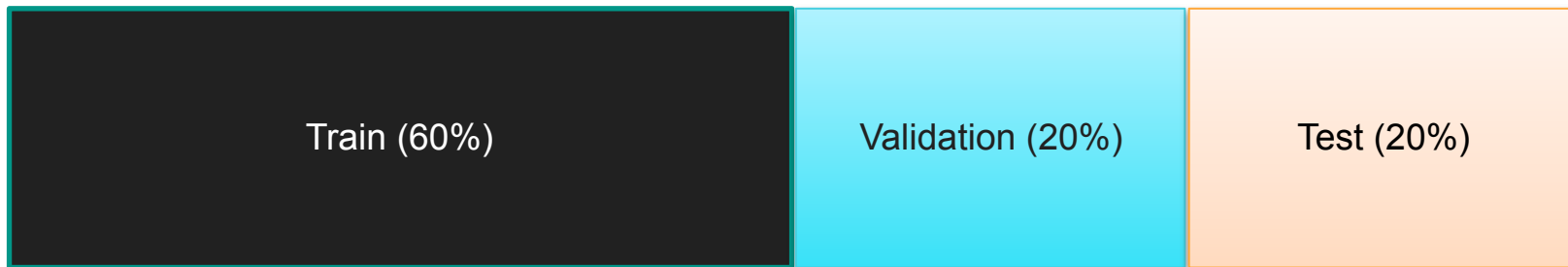
# Validation And Testing: **Selecting Models And Estimating Their Quality**

---

# LEARNING GOALS



1. Understand the importance of estimating model generalization error (testing) and practical methodologies for doing so
2. Learn validation/cross-validation strategies for model selection
3. Recognize the difference between validation and testing



---

# Testing: Estimating Generalization Error

---

METIS



**Generalization Error:** How well can we expect a model to perform on new data from the same distribution as the training data?

- Predictive models are only *useful* if they can give us good target approximations for samples that we haven't seen before
- Example: Zillow predicts the market value of a home before it's listed for sale, training a model on known listing prices
- So when evaluating models, we should attempt to measure how well they *generalize*, i.e. estimate performance on samples we didn't train on. We call this **testing**.

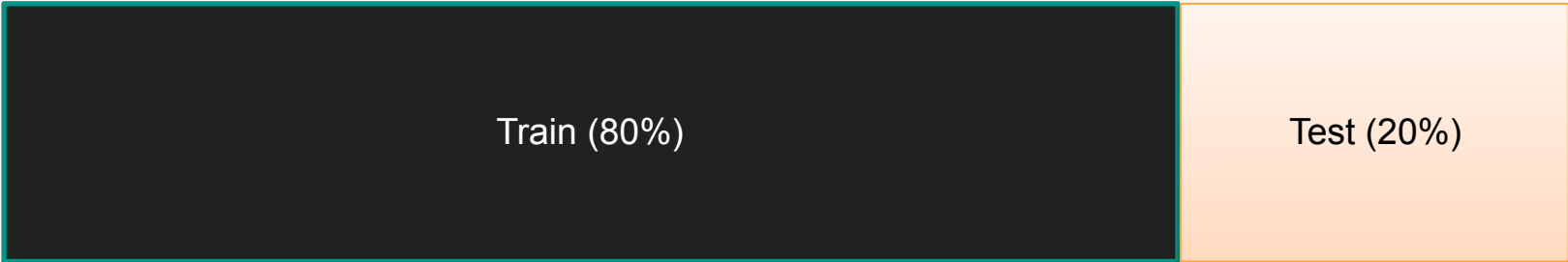
# TESTING, IN PRACTICE



**Simulate generalization:** We can *hold out* a portion of our labeled dataset to simulate the real-world challenge of unseen samples

We call this a **test set**, and exclude it from the data we train on

We then *estimate generalization error* by making predictions on test, and scoring those predictions against the ground truth (our test labels)



Train (80%)

Test (20%)

# TESTING, IN PRACTICE; cont.



## 1. Fit model to training data



Train (80%)

Test (20%)

# TESTING, IN PRACTICE; cont.



1. Fit model to training data
2. Score model on testing data



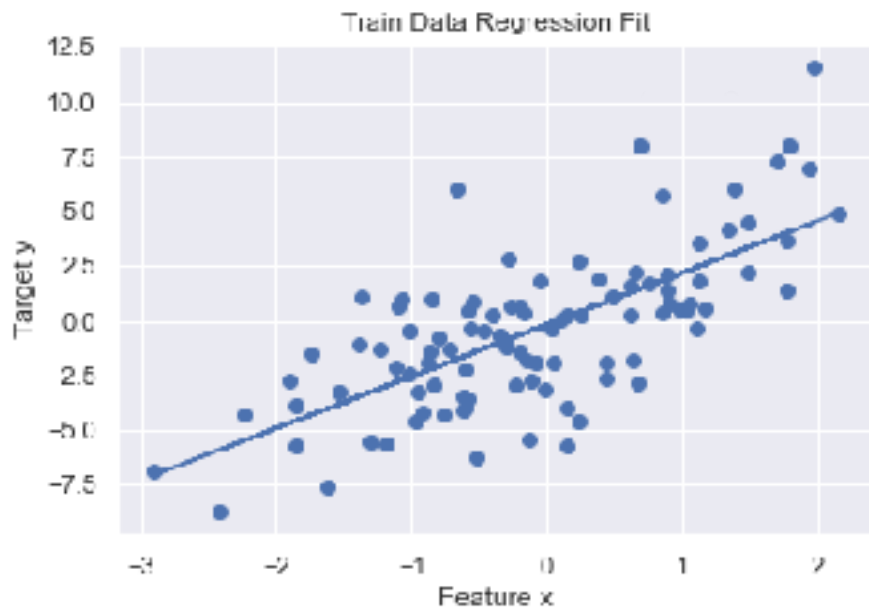
Train (80%)

Test (20%)

# TEST USUALLY UNDERPERFORMS TRAIN



**Model is optimized to perform as well as possible on train**, so it's no surprise that it tends to have a worse evaluation score on test (though this is not guaranteed).



→  $R^2: .49$



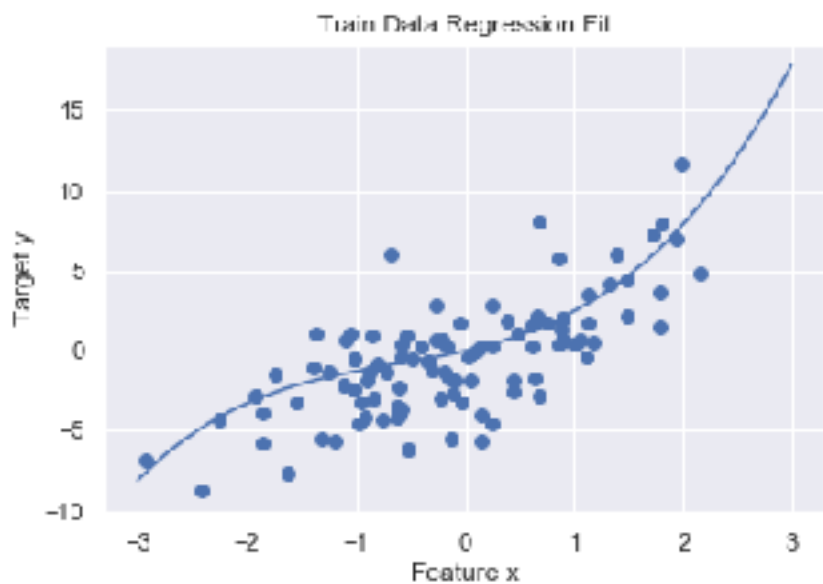
→  $R^2: .43$



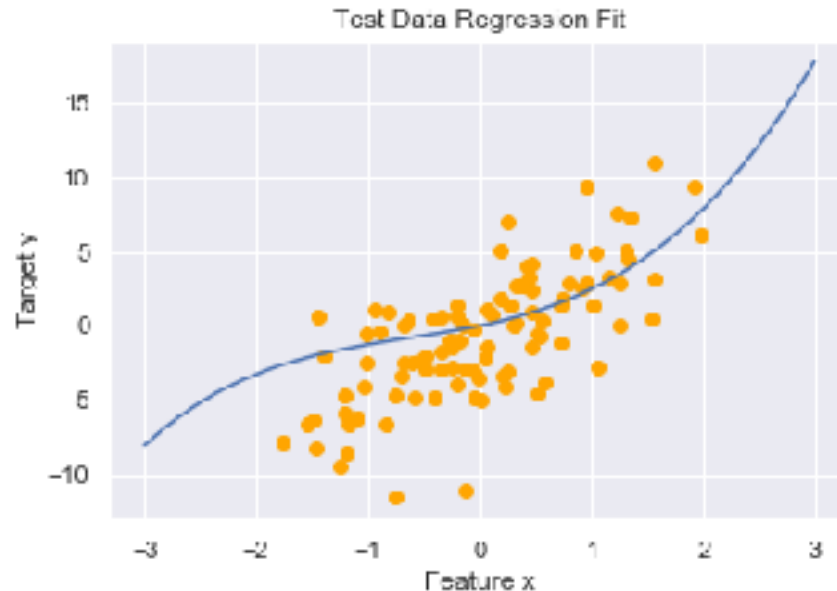
# THE RISK OF OVERFITTING



**Out of sample evaluations let us check for overfitting:** more complex models can get arbitrarily better at predicting the train data, but will start to fit to spurious patterns and generalize more poorly



→ Degree 4  $R^2$ : .54



→ Degree 4  $R^2$ : .42

---

# Validation: Optimizing Our Modeling Choices

---

METIS

# VALIDATION: OPTIMIZING CHOICES



When we construct predictive models, we typically have **many choices**:

- Features: which data columns do we include/exclude or engineer?
- Preprocessing: how should we handle nulls? Should we standardized the scale of the features?
- Hyper-parameters: What degree polynomial regression should we fit? What regularization strength should we use? How does a random forest model compare to a linear regression model?

# VALIDATION IN PRACTICE



We can make some choices using our domain knowledge and good instincts, but a **validation framework** gives us an empirical way to choose and avoid over/under-fitting

We validate with the usual *best generalization* end-goal in mind: we exclude validation data from training, and use it to score predictions across a range of model choices

We can then *select* a choice of model based on the strongest validation score - i.e., this score gives us **direct feedback on a possible choice**. Once we've chosen a model, we can combine our train and validation sets, retrain the model, and get the test score

Train (60%)

Validation (20%)

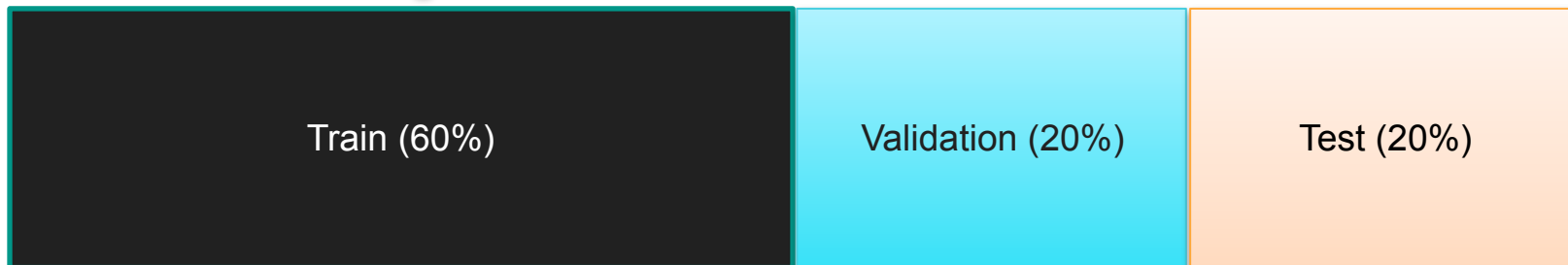
Test (20%)

# VALIDATION IN PRACTICE; cont.



## 1. Train candidate models

- Linear Regression
- Polynomial Regression
- Ridge Regression



# VALIDATION IN PRACTICE; cont.



## 1. Train candidate models

- Linear Regression
- Polynomial Regression
- Ridge Regression

## 2. Score candidates

→  $R^2$ : .35

→  $R^2$ : .50

→  $R^2$ : .25

Train (60%)

Validation (20%)

Test (20%)



# VALIDATION IN PRACTICE; cont.



## 3. Retrain best candidate on train + validation

→ Polynomial Regression



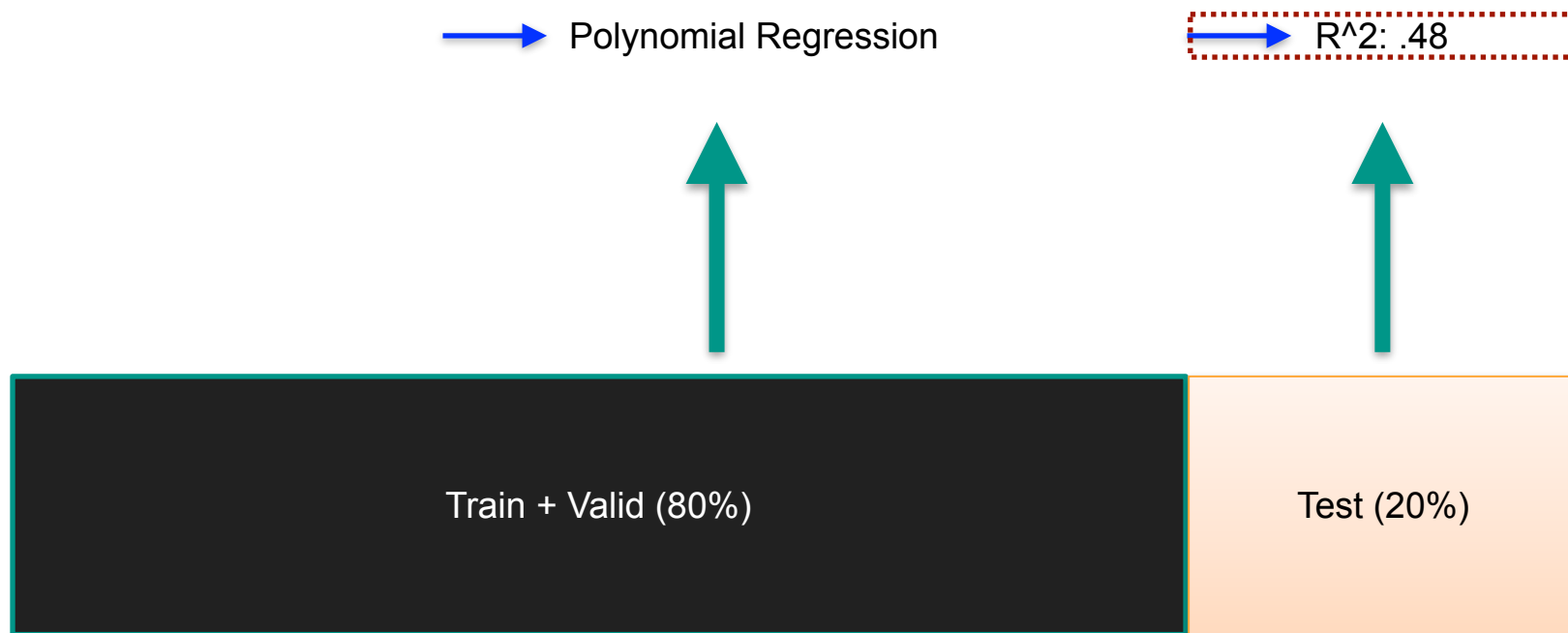
Train + Valid (80%)

Test (20%)

# VALIDATION IN PRACTICE; cont.



## 4. Score final model on test





# VALIDATION: KEY CONSIDERATIONS



**Validation is not testing:** This is a very common pitfall. Once you've used a data set to influence your model choices through direct feedback, it can't be used to give an unbiased estimate of generalization error

**Fair comparisons:** Candidate models should be compared against the same validation scheme

**Data efficiency:** after we use a portion of our data for validation, we should reuse it as training data to improve the final model

---

# Cross Validation: Optimizing Our Optimization Of Choices

---

METIS

# CROSS VALIDATION: ADDING MORE RIGOR



**Cross-validation is about reliability and efficiency:** what if we overfit to an unlucky validation set? Can we use more of our data than just one hold-out for validation?

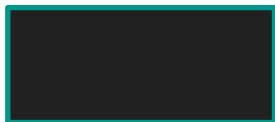
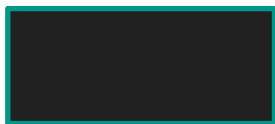
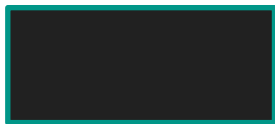
**K-Fold partitioning:** randomly divide our non-test data into  $K$  equal-sized groups. Each group will be used as a validation set once, and we'll compare candidate models via mean scores across all validation scores.

**K is usually 5 or 10:** depends on problem and size of data, but these are common choices

# CROSS VALIDATION IN PRACTICE



Train data:  
divided into  
5 folds



Do the following,  
for each  
candidate model

Test, held out

# CROSS VALIDATION IN PRACTICE

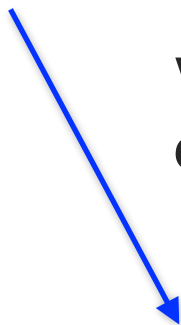


Train data:  
divided into  
5 folds



**Train** on  
black folds

**Validate** on  
cyan fold

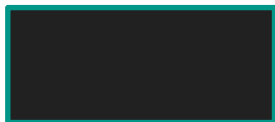
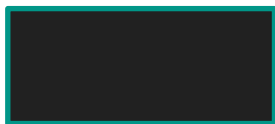
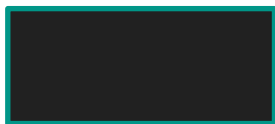
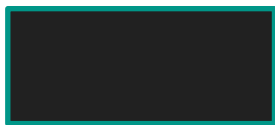


Validation  $R^2$  scores: .45,

# CROSS VALIDATION IN PRACTICE



Train data:  
divided into  
5 folds



**Train** on  
black folds

**Validate** on  
cyan fold

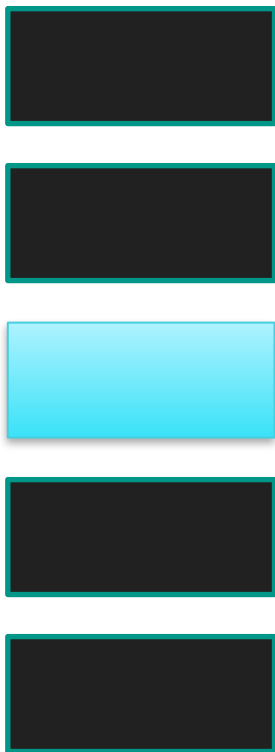


**Validation  $R^2$  scores:** .45, .39,

# CROSS VALIDATION IN PRACTICE



Train data:  
divided into  
5 folds



**Train** on  
black folds

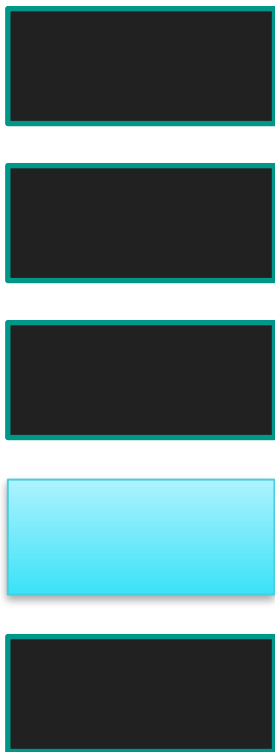
**Validate** on  
cyan fold

**Validation  $R^2$  scores:** .45, .39, .57,

# CROSS VALIDATION IN PRACTICE



Train data:  
divided into  
5 folds



**Train** on  
black folds

**Validate** on  
cyan fold

**Validation  $R^2$  scores:** .45, .39, .57, .49,



# CROSS VALIDATION IN PRACTICE

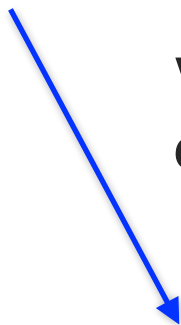


Train data:  
divided into  
5 folds



**Train** on  
black folds

**Validate** on  
cyan fold

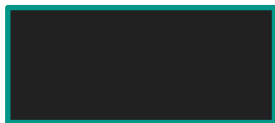
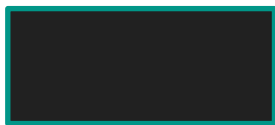
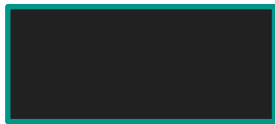
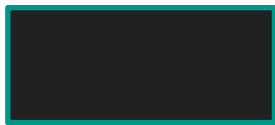


**Validation  $R^2$  scores:** .45, .39, .57, .49, .50

# CROSS VALIDATION IN PRACTICE



Train data:  
divided into  
5 folds



Produces a set of results for each  
candidate model

Linear regression



Validation  $R^2$  scores: .45, .39, .57, .49, .50

Poly regression

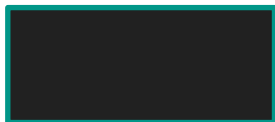
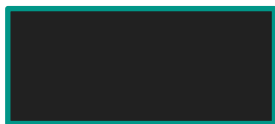
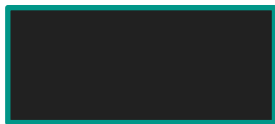


Validation  $R^2$  scores: .53, .43, .67, .55, .51

# CROSS VALIDATION IN PRACTICE



Train data:  
divided into  
5 folds



Summarize candidates by mean  
score, select best

Linear regression



5-Fold validation mean  $R^2$  score: .48

Poly regression

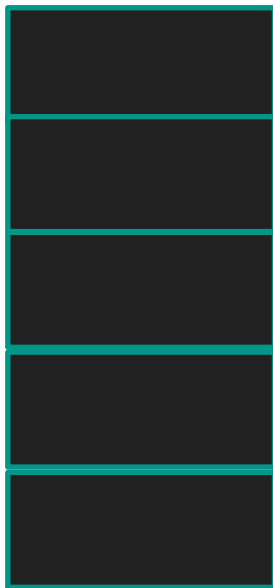


5-Fold validation mean  $R^2$  score: .54

# CROSS VALIDATION IN PRACTICE



Train data:  
recombined



Polynomial regression  
selected as best  
candidate model

**Poly regression,  
retrained on all data,  
final score on test**



Test, held out

$R^2$ : .48

---

# Validation And Testing: Recap

---

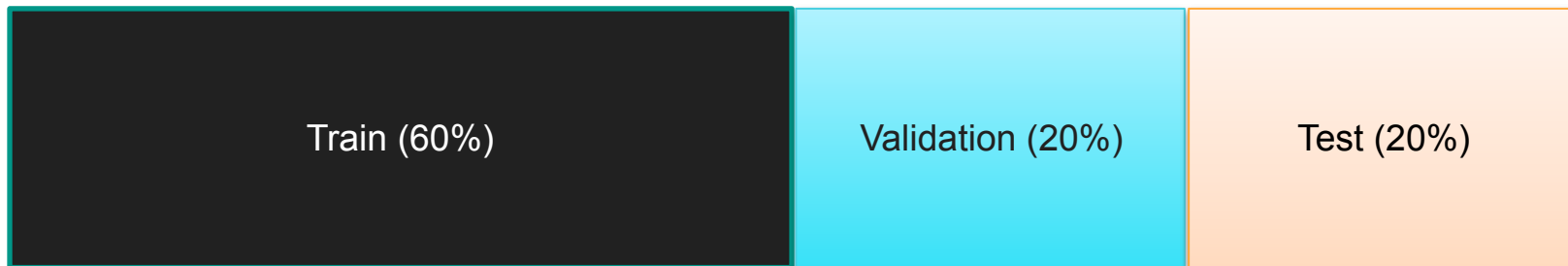
METIS

# WORKFLOW METHOD 1: Train/Valid/Test



Collect set of candidate models. Fit each on train, score on validation, select final model via best validation score

Retrain final model on train + validation, report score on test as estimate of generalization error



# WORKFLOW METHOD 2: CV/Test



Collect set of candidate models. Run each through a K-fold CV loop, select final model via best mean validation score

Retrain final model on combined folds, report score on test as estimate of generalization error



# VALIDATION VS. CV - WHEN TO USE?



**Simple validation** is significantly faster and often representative enough when working with very large samples (~millions+)

**Cross validation** is more appropriate with small-medium size data or when variance in results between different validation sets is high



# CAN WE PUSH EVEN FURTHER?



There's nothing stopping us from doing **repeated rounds of CV** with different random K-folds for even more rigor

- Also an alternate form of testing: run CV on all data to select, then run another K-Fold loop to get multiple/mean out of sample scores

We usually take means across validation folds in CV to compare model candidates, but we can also gain information from **distributions of scores across folds** (e.g. variance)

**Nested CV** is another advanced technique for a more robust combination of CV and testing: [see here for detail](#).

# SUMMARY



## Training

In sample

Model building

Optimize model  
parameters (fit)

## Validation

Out of sample

Feedback to model  
selection

Optimize choices:  
features and model  
hyper-parameters

## Testing

Out of sample

No feedback to model  
selection

Final estimate of model  
generalization error

A close-up photograph of a Shiba Inu dog's face, looking slightly to the left with wide, dark eyes. The dog has light tan fur with darker markings around its eyes and on its ears. The background is a soft, out-of-focus yellow. Three lines of white, bold, sans-serif text with black outlines are overlaid on the image. A small watermark is visible in the bottom left corner.

**VALIDATION SO WOW**

**SUCH MODEL SELECTION**

**VERY NOT ESTIMATE  
OF GENERALIZATION ERROR**