Data Splitting

February 26, 2026

Announcements

- Thank you for filling out the mid-semester feedback survey
  - I will continue to leave this form open in case you would like to provide anonymous feedback throughout the rest of the semester
- Based on feedback, deadlines are now **11:59pm**

Today's plan

1. **Review** of last time
2. **More on Data Splitting**
3. **Introduction to Lab 3**

Last Time: Assessing Generalizability

To rigorously assess generalizability of a model, we need two things:

- **Metrics** to measure generalization error
  - Regression: RMSE, MSE, R^2, MAE, ...
  - Classification: Accuracy, F1, AUROC, AUPRC, ...
- **Data** to measure generalization error
  - Different choices for how to do data splitting: random sampling, group-wise sampling, time-based sampling, spatial-based sampling...
    - Python: check out sklearn data splitting guide
    - R: check out caret data splitting guide
  - Data splitting should mimic how we expect to obtain our future data
  - If we do not perform data splitting properly, our test error will most likely be too optimistic

Data Splitting

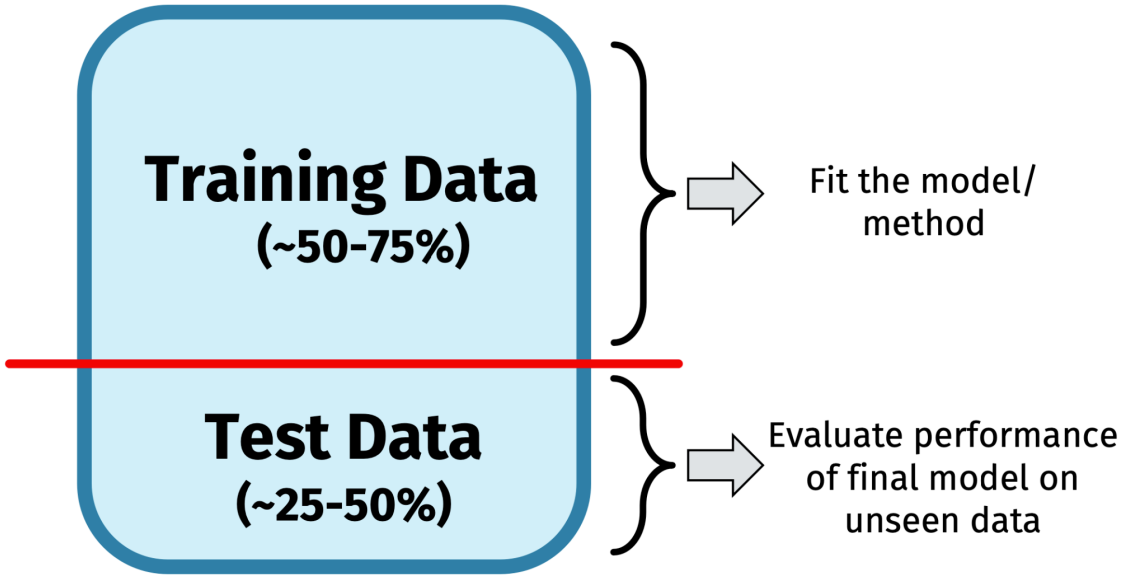
Overview of Data Splitting

Data splitting is the key to assessing generalizability (or how well our method performs on future unseen data)

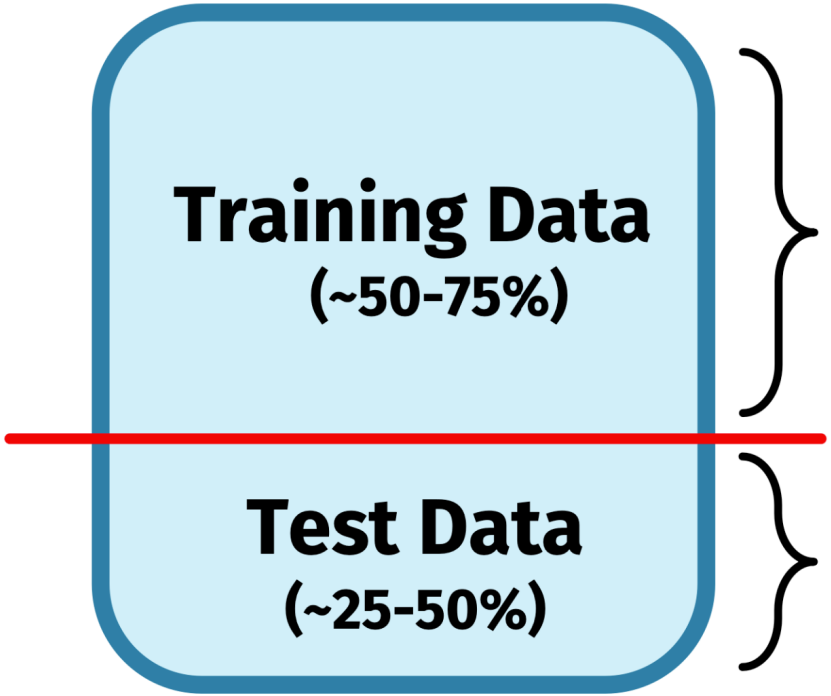
The simplest case:

(ignoring choice of hyperparameters and possibility of multiple models)

Q: How should we allocate samples to the training data versus the test data?



Data splitting with hyperparameter tuning

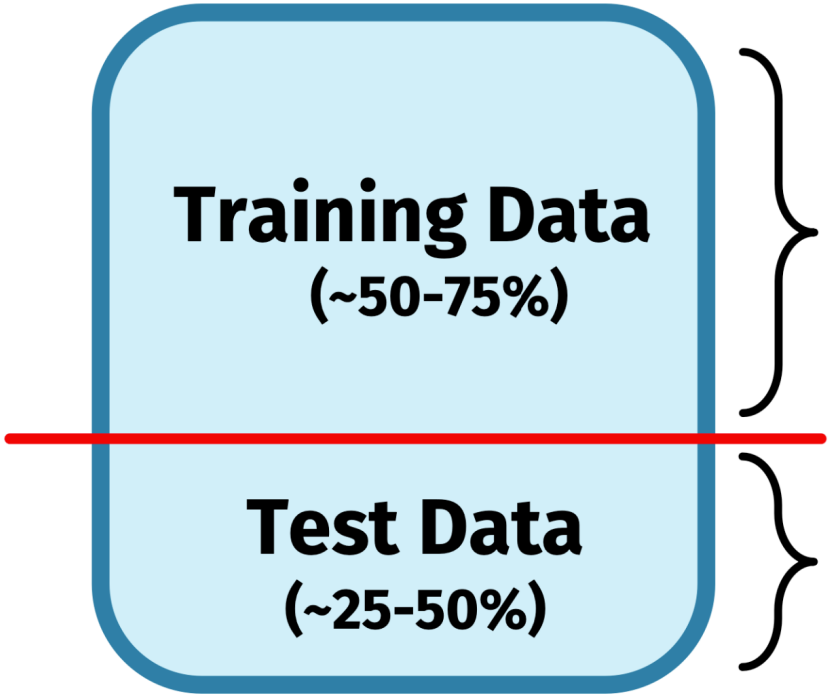


Idea: for each possible hyperparameter candidate,

- Split training data into "training" & "test"
- Select hyperparameter that minimizes error on test set

Evaluate performance of final model on unseen data

Data splitting with hyperparameter tuning

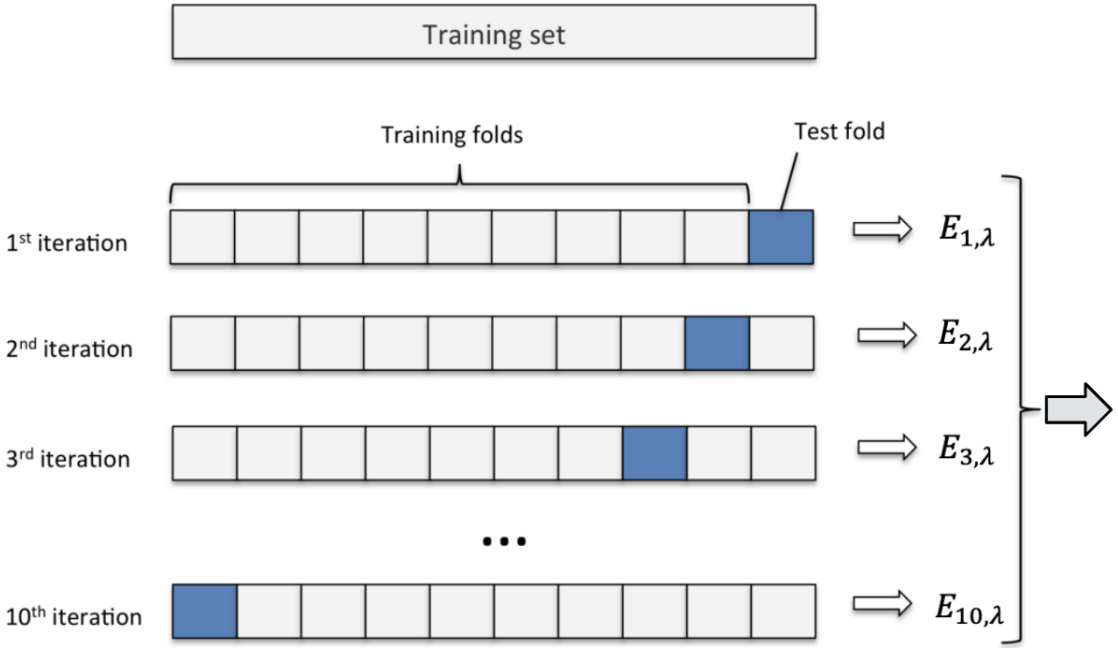


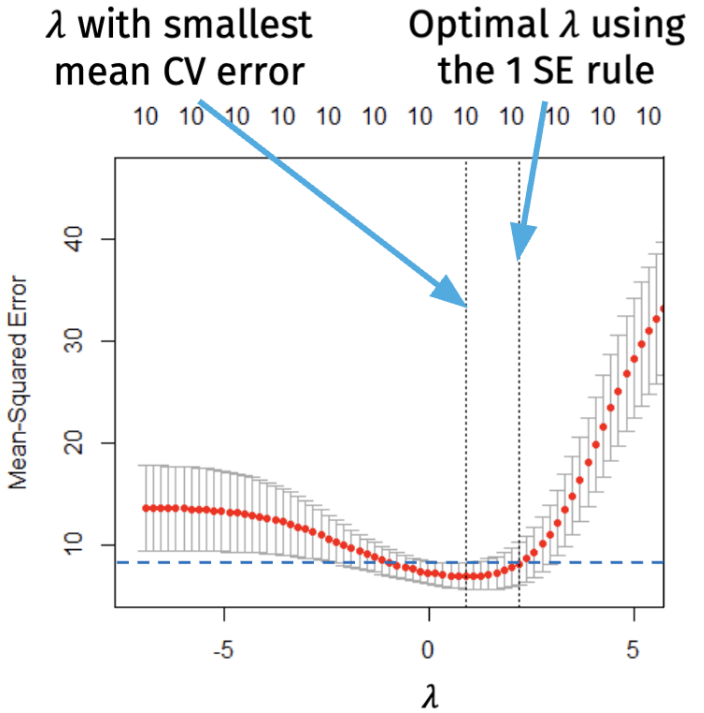
Idea: for each possible hyperparameter candidate,

- Split training data into "training" & "test"
- Select hyperparameter that minimizes error on test set

Evaluate performance of final model on unseen data

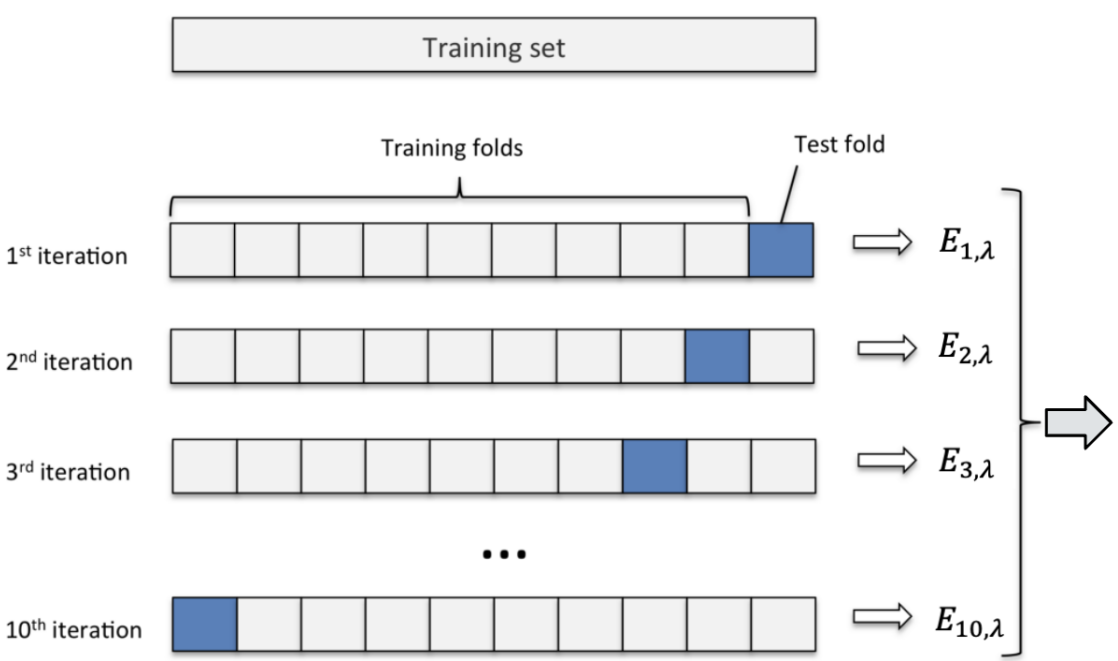
K-fold Cross-Validation (CV) for choosing hyperparameters

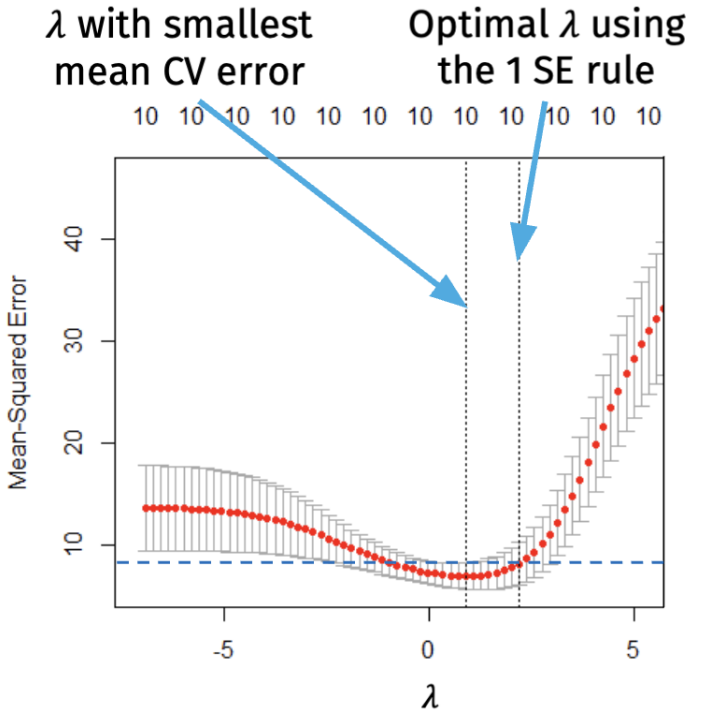




Advantage of CV over repeated data splitting: minimum amount of computation needed while still ensuring every sample is in the test set exactly one time

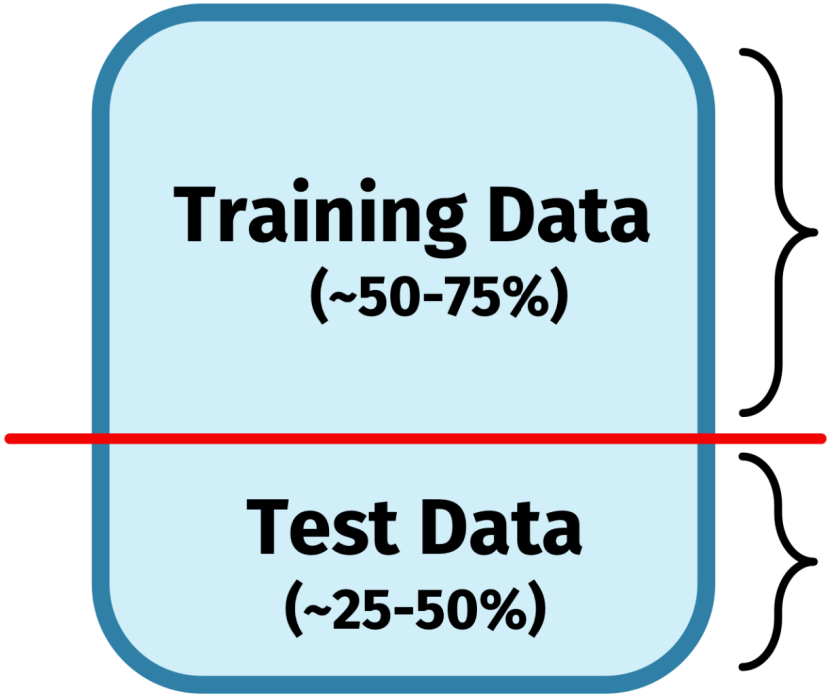
K-fold Cross-Validation (CV) for choosing hyperparameters





How do we split the data into folds? Typically similar to how we split into train vs test

Data splitting with hyperparameter tuning

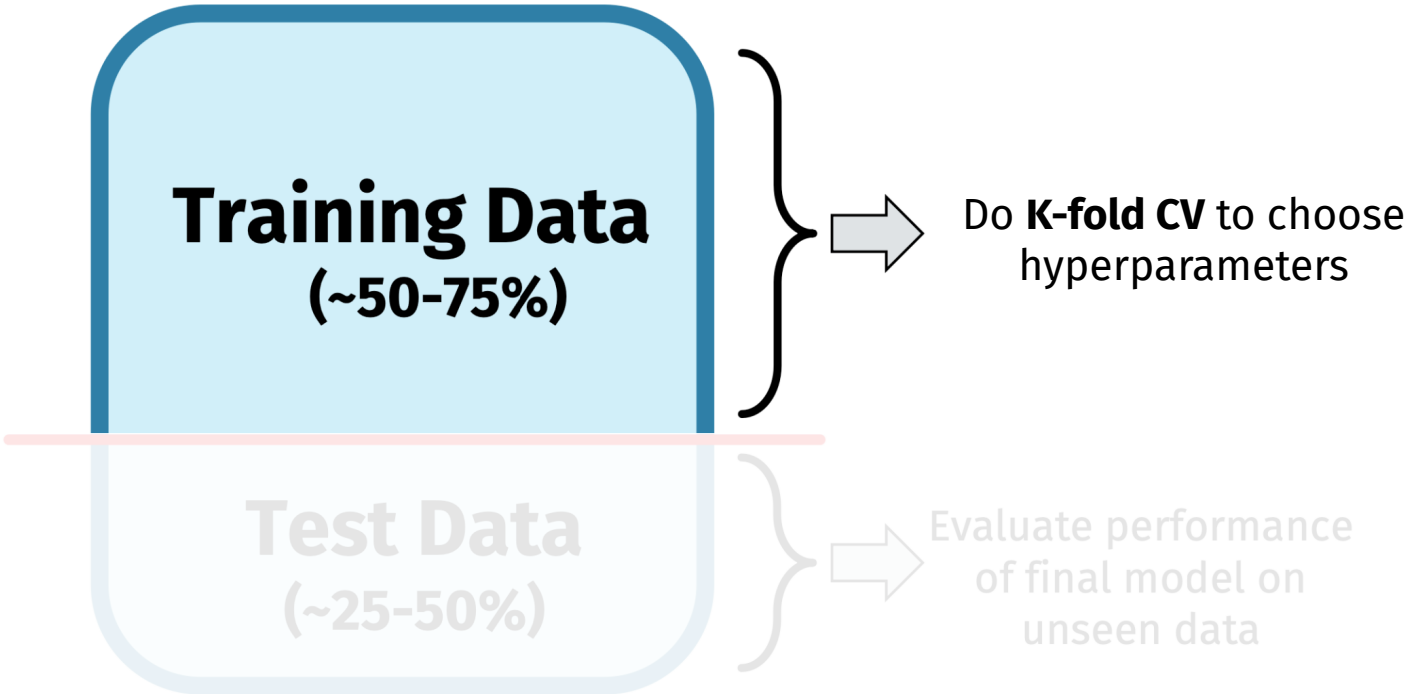


1. Do **K-fold CV** to choose hyperparameters
2. Re-fit model on full training data using optimal hyperparameters

Evaluate performance of final model on unseen data

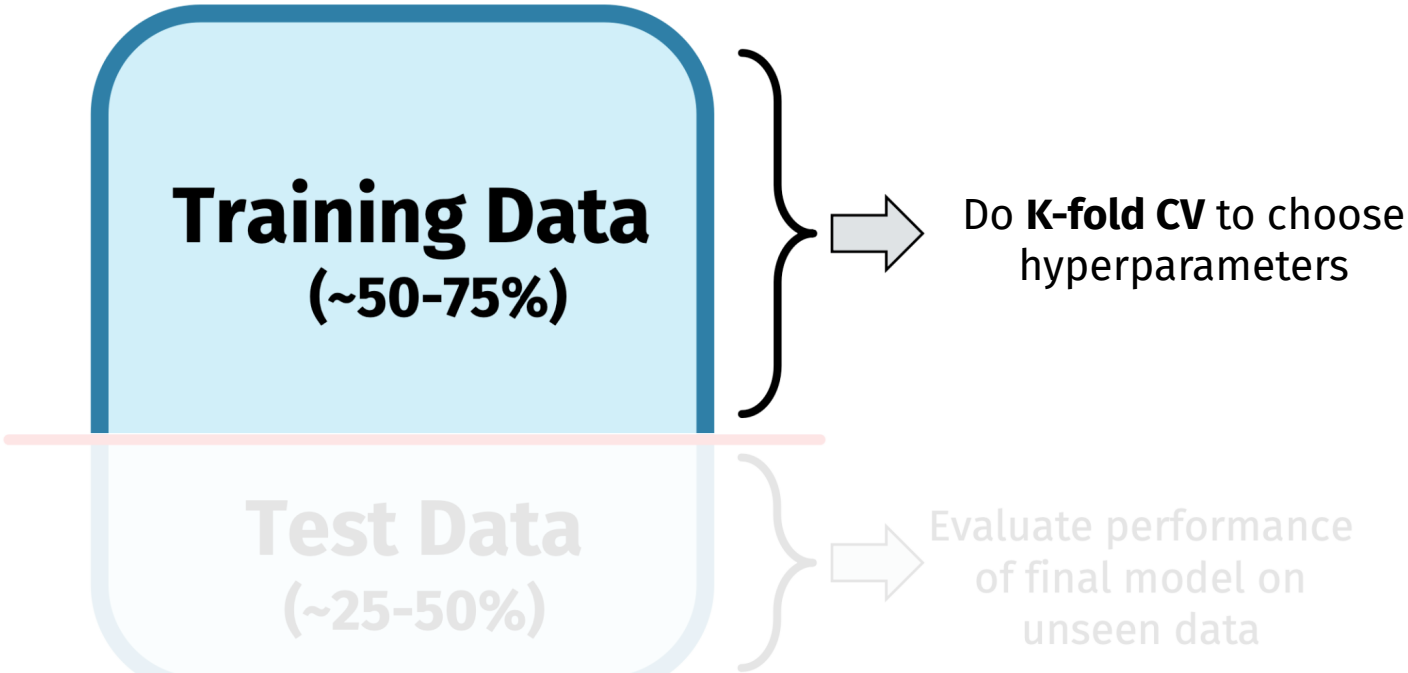
Data splitting with hyperparameter tuning

Caution: Do **NOT** report the CV error for the best hyperparameter as our test error!



Data splitting with hyperparameter tuning

Caution: Do **NOT** report the CV error for the best hyperparameter as our test error!

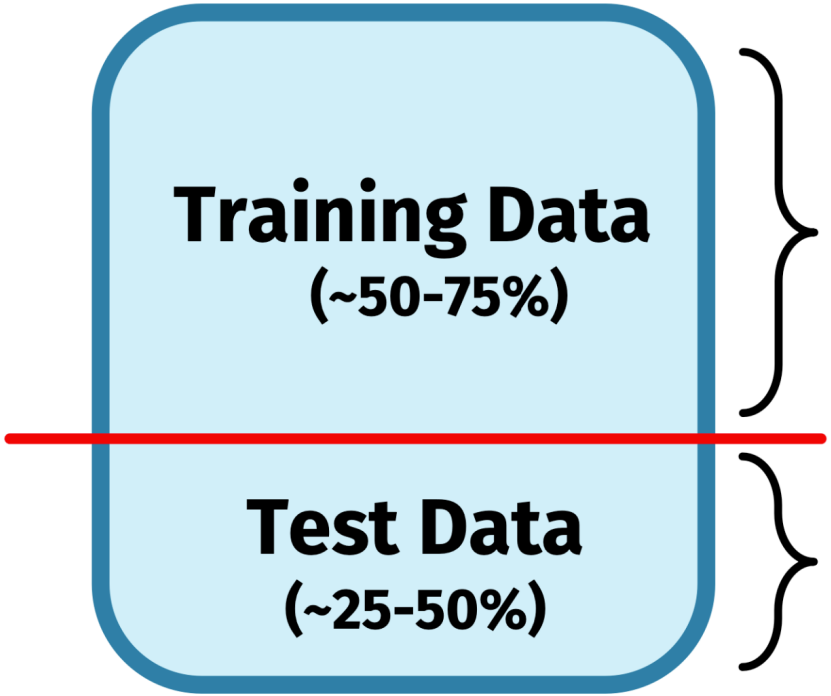


This would be an **overly-optimistic estimate** because we essentially took the minimum (or best) error across many attempts ("too good to be true")

Data splitting with hyperparameter tuning + model selection

(multiple models to choose from)

What if we report the test error after using test data to do model selection?



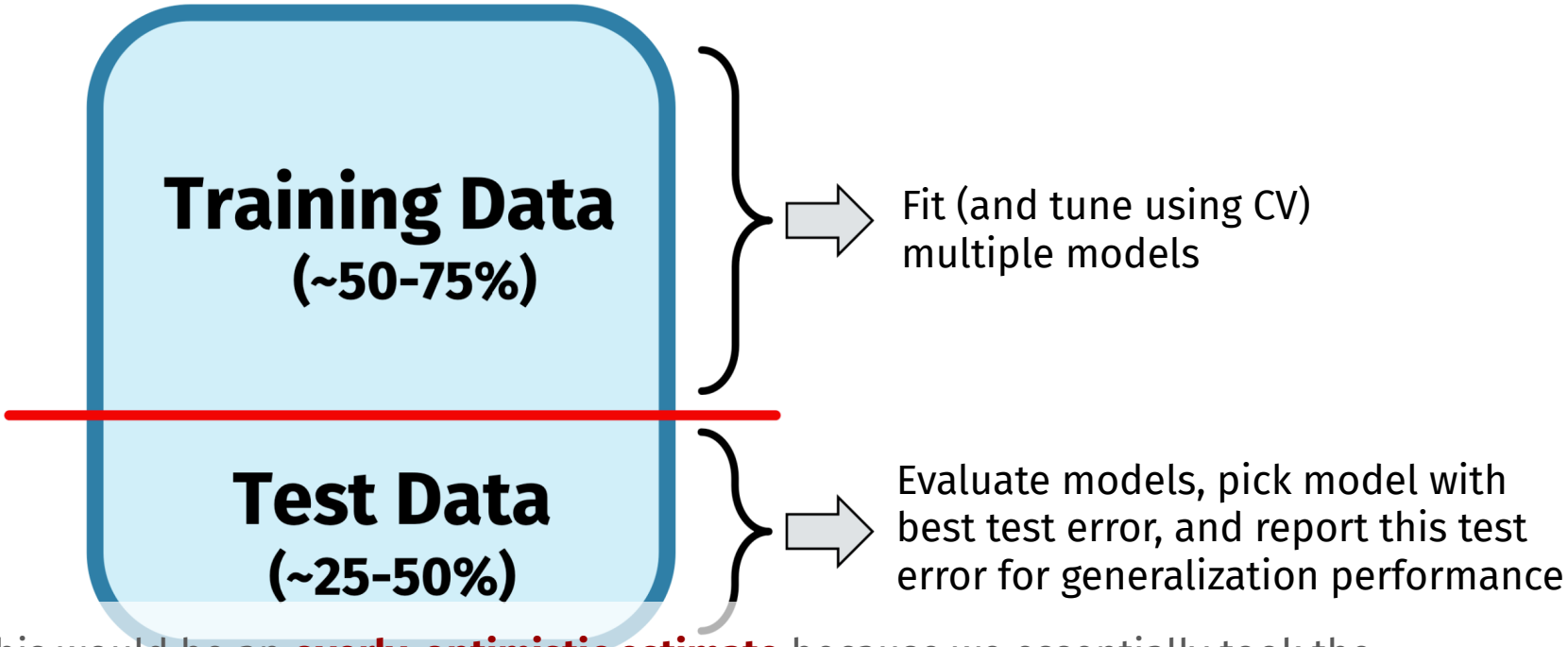
Fit (and tune using CV) multiple models

Evaluate models, pick model with best test error, and report this test error for generalization performance

Data splitting with hyperparameter tuning + model selection

(multiple models to choose from)

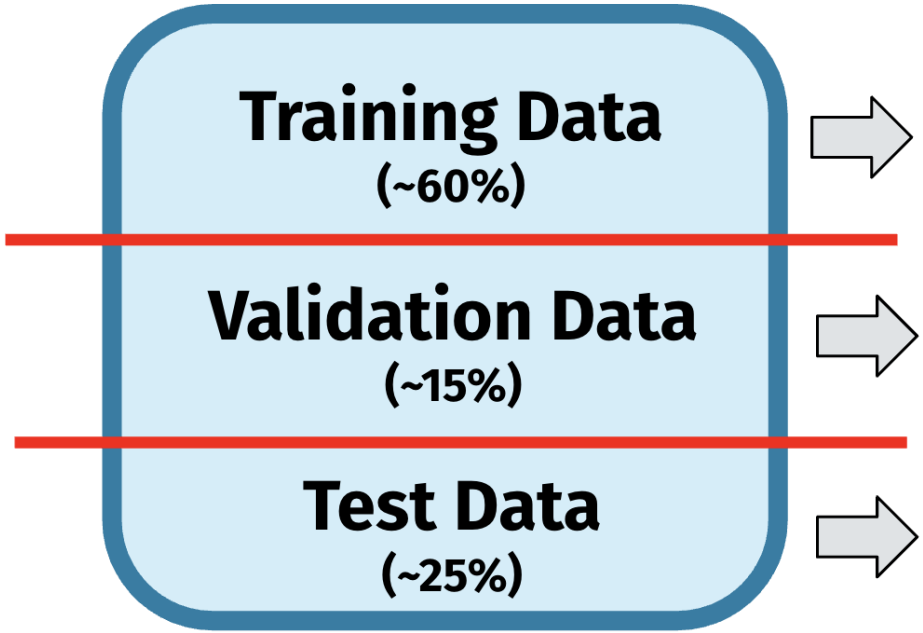
What if we report the test error after using test data to do model selection?



Again, this would be an **overly-optimistic estimate** because we essentially took the minimum (or best) error across many attempts ("too good to be true")

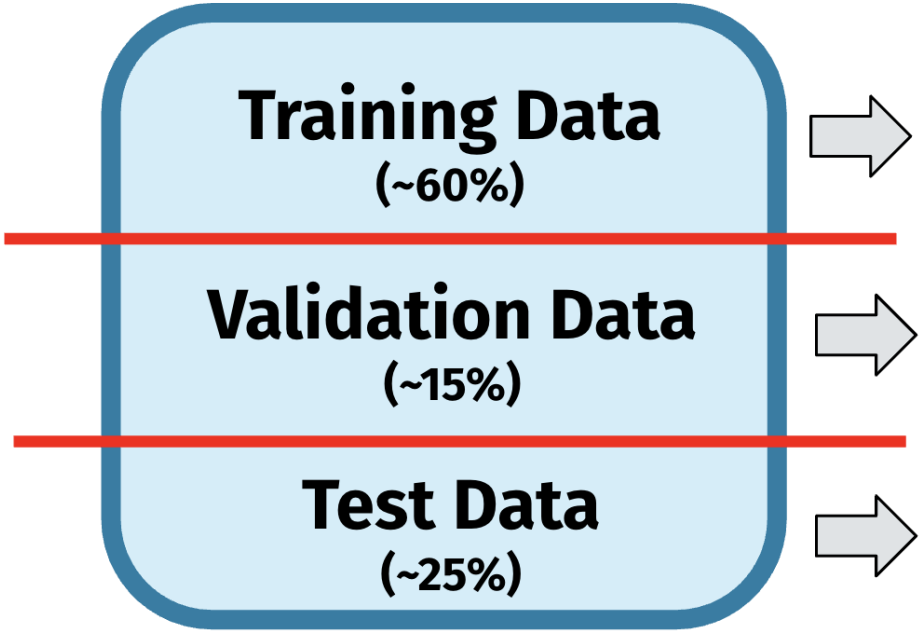
Data splitting with hyperparameter tuning + model selection

(multiple models to choose from)



Data splitting with hyperparameter tuning + model selection

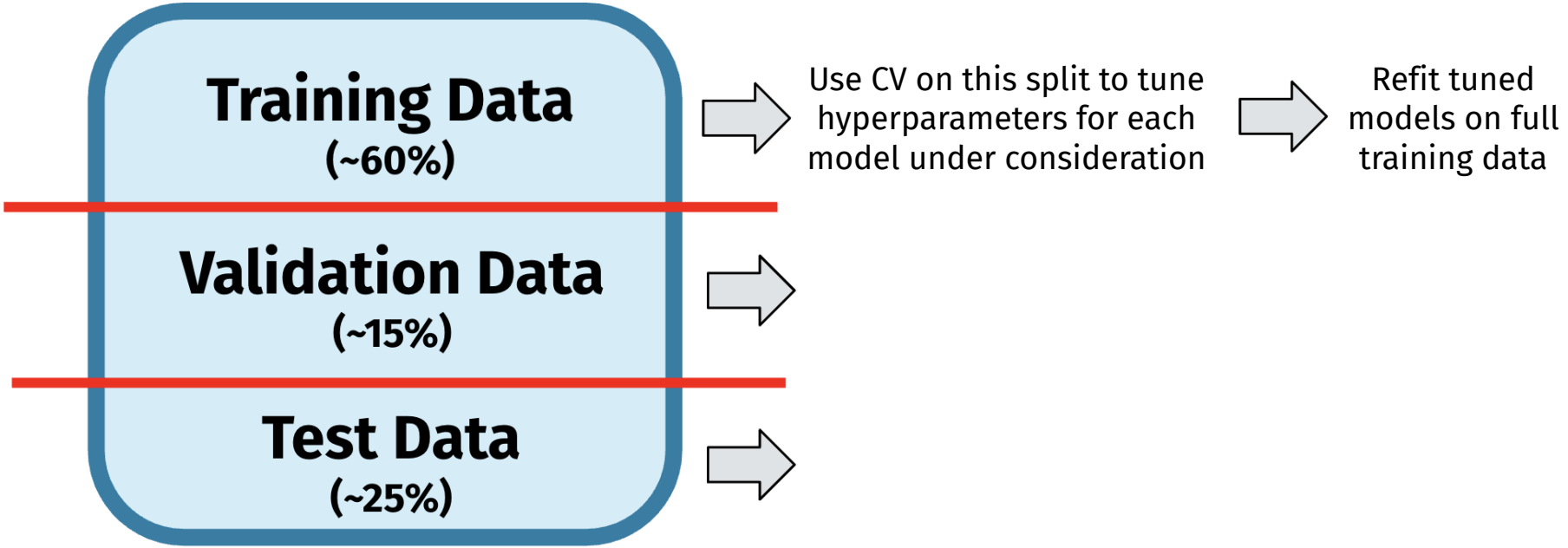
(multiple models to choose from)



Use CV on this split to tune hyperparameters for each model under consideration

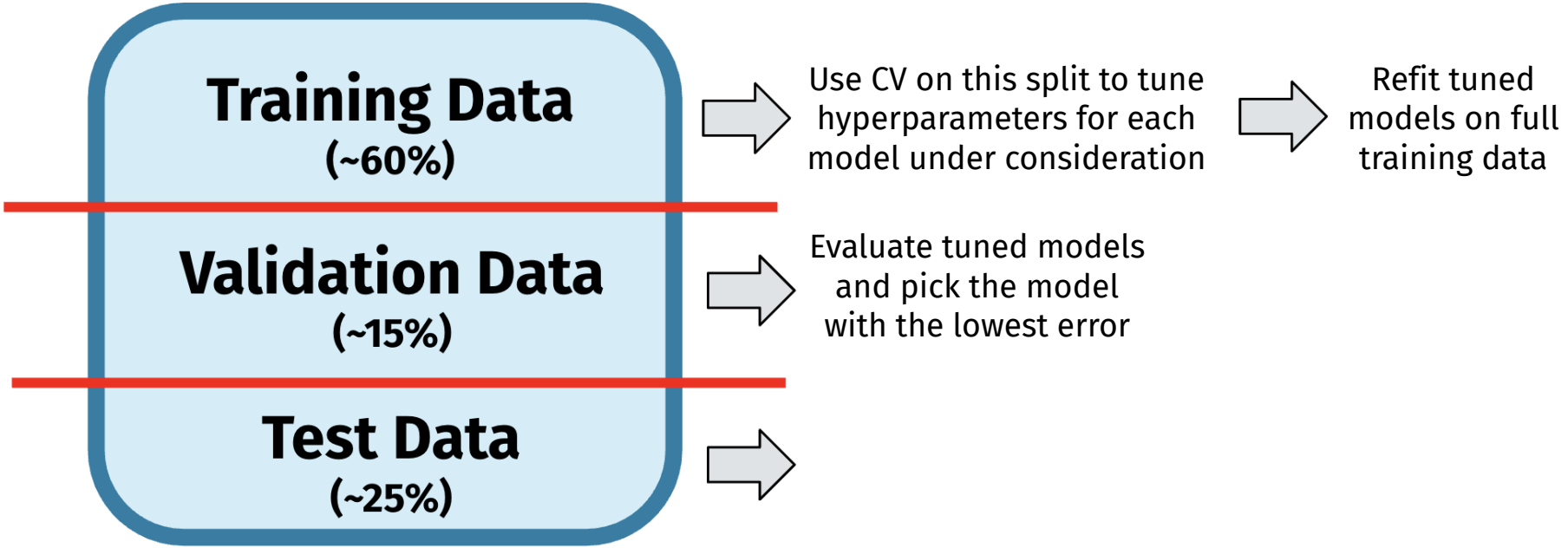
Data splitting with hyperparameter tuning + model selection

(multiple models to choose from)



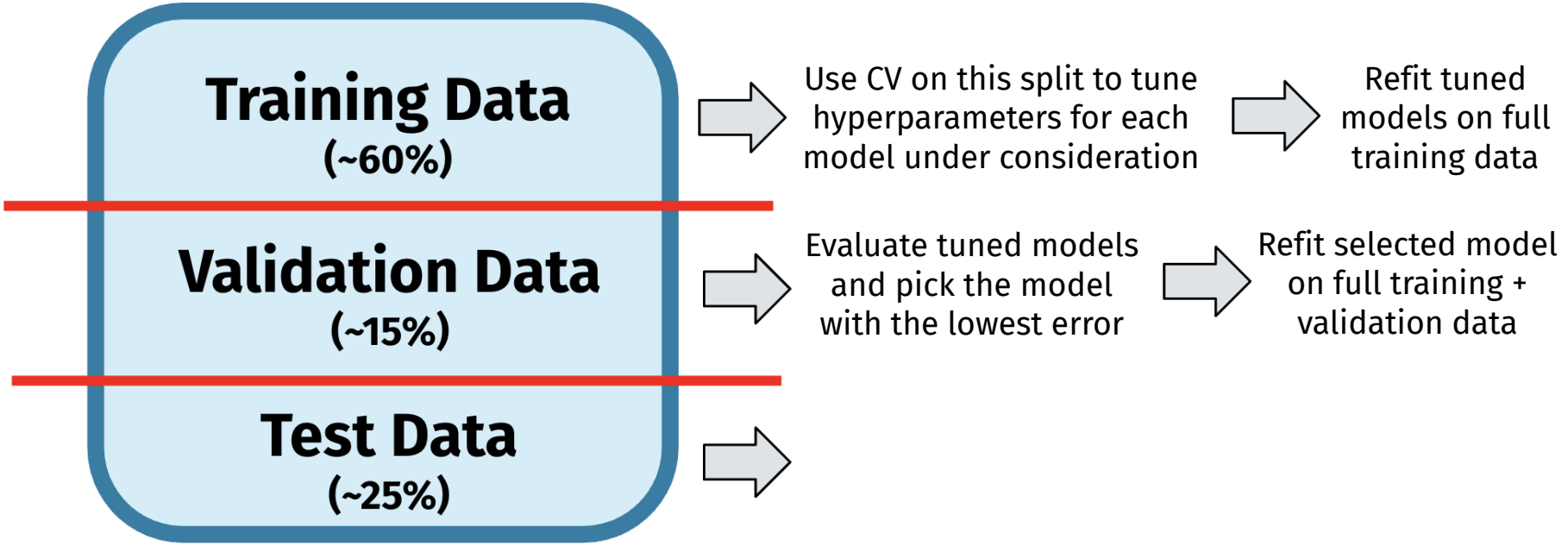
Data splitting with hyperparameter tuning + model selection

(multiple models to choose from)



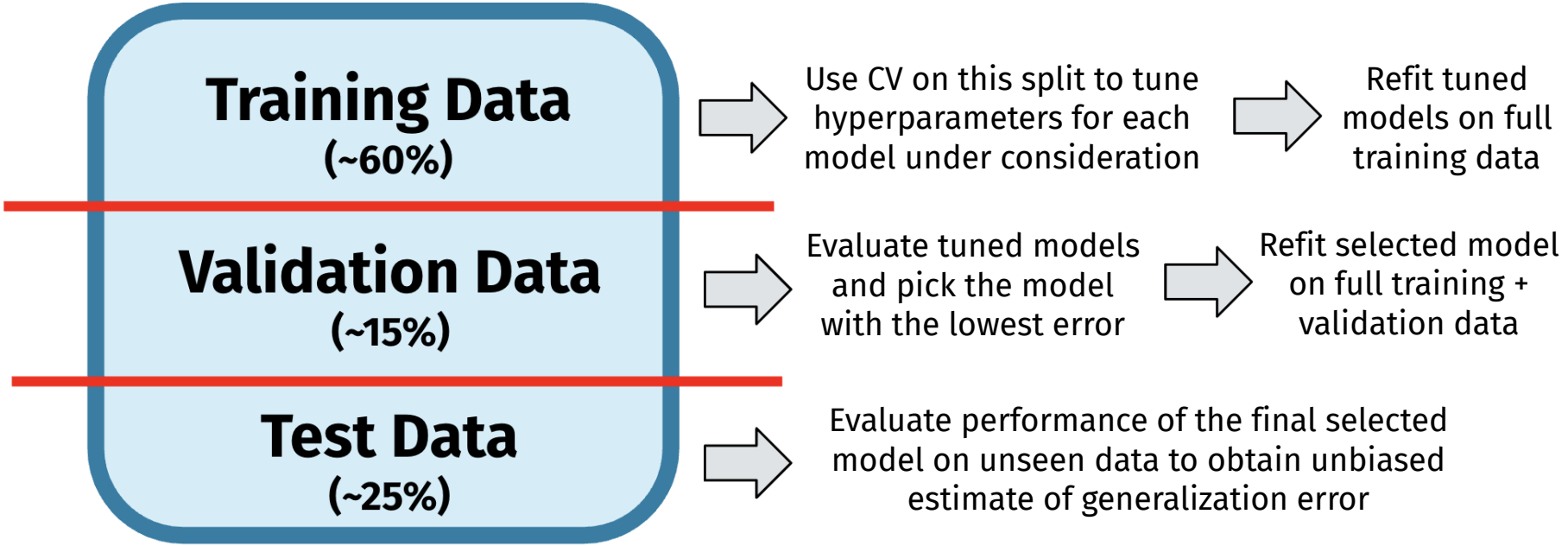
Data splitting with hyperparameter tuning + model selection

(multiple models to choose from)



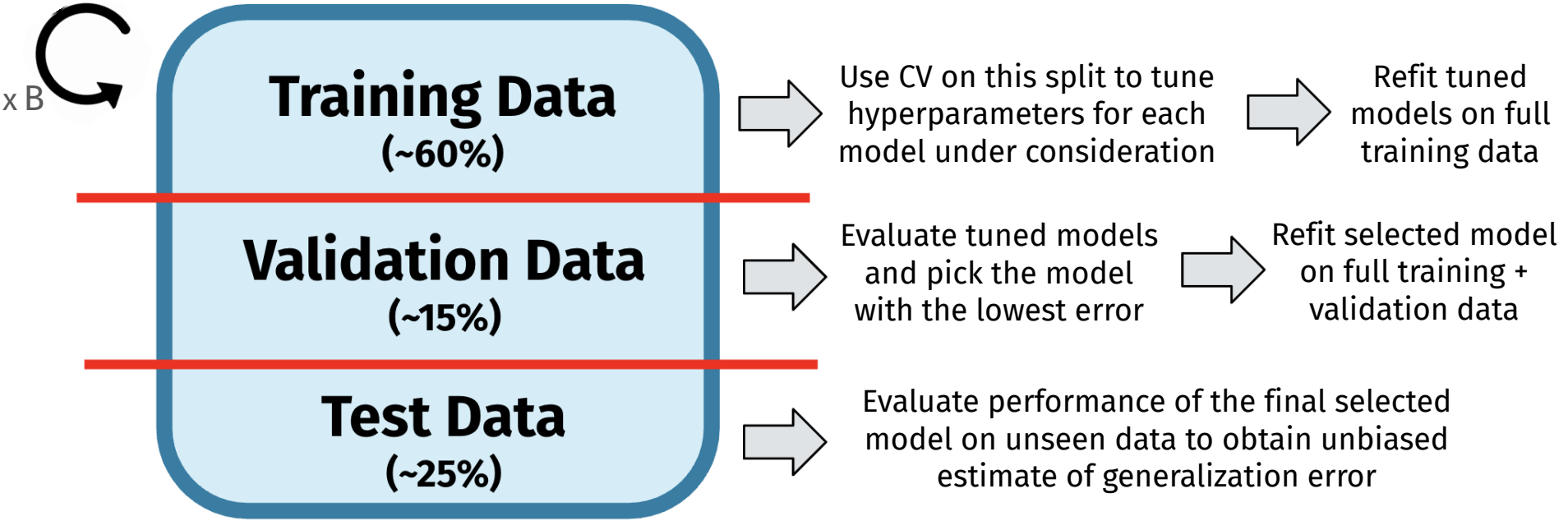
Data splitting with hyperparameter tuning + model selection

(multiple models to choose from)



Data splitting with hyperparameter tuning + model selection

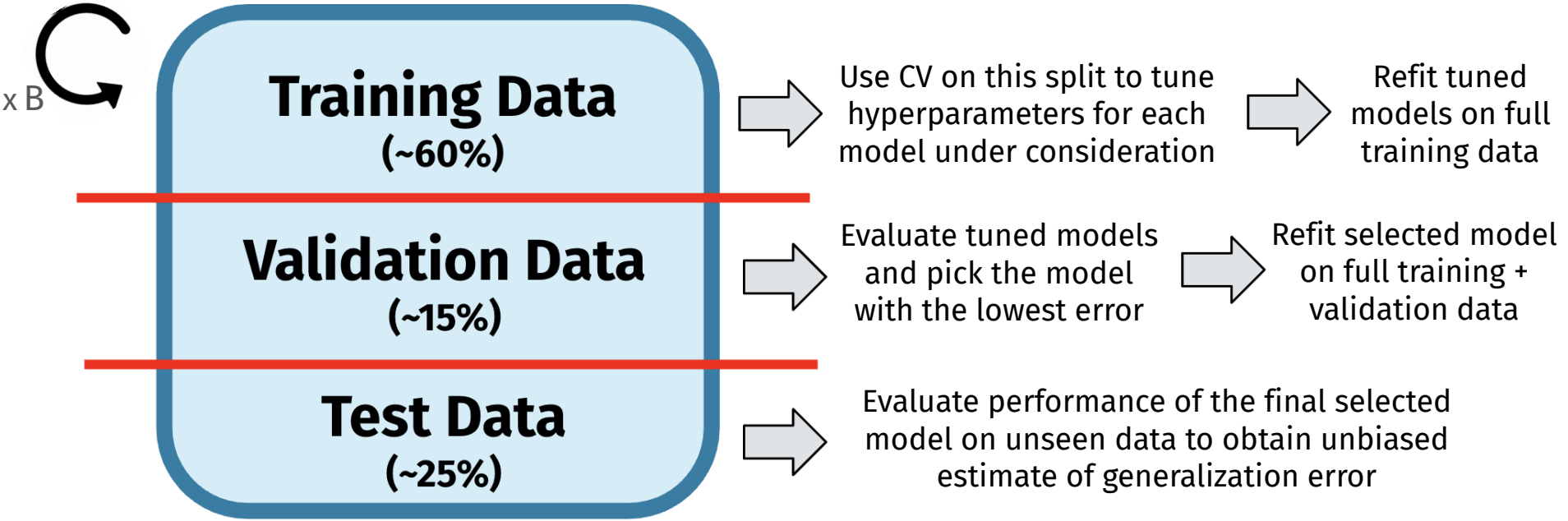
(multiple models to choose from)



+ Repeat this data splitting B times to get a variance estimate of the test error

Data splitting with hyperparameter tuning + model selection

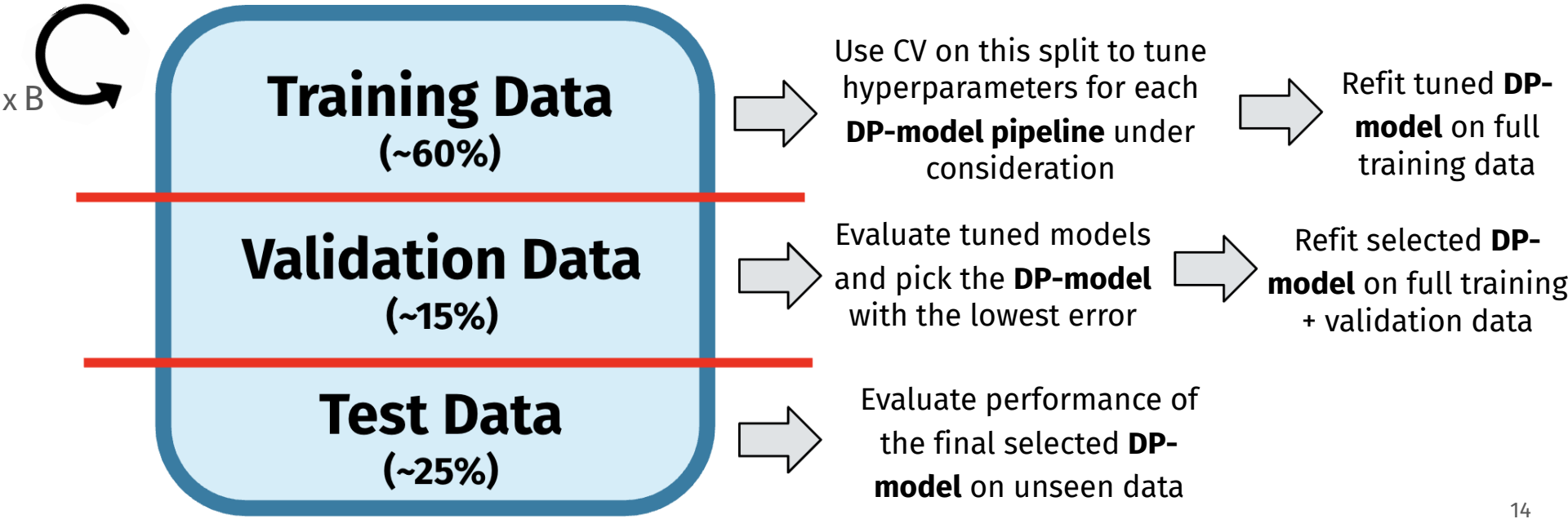
(multiple models to choose from)



- Repeat this data splitting B times to get a variance estimate of the test error
- This gives you an unbiased estimate of the prediction error for the **statistical learning pipeline/process**, **NOT a specific model**

Data splitting with hyperparameter tuning + DP-model selection

How do we do data splitting when we have **multiple reasonable data preprocessing (DP)** pipelines?



Data splitting with hyperparameter tuning + DP-model selection

What if we want to build a March Madness prediction model?

- Considering multiple DP-model pipelines
- Data from 10 tournaments/years
- Want to generalize to new tournaments/years

Example data splitting algorithm

Algorithm 1: Example using Repeated Data Splitting

2 Split data $\mathcal{D}$ into training $\mathcal{D}^{\text{train}}$, validation $\mathcal{D}^{\text{valid}}$, and test $\mathcal{D}^{\text{test}}$
[e.g., 6 tournaments in $\mathcal{D}^{train}$, 2 tournaments in $\mathcal{D}^{valid}$, and 2 tournaments in $\mathcal{D}^{test}$]

Example data splitting algorithm

Algorithm 1: Example using Repeated Data Splitting

Split data $\mathcal{D}$ into training $\mathcal{D}^{\text{train}}$, validation $\mathcal{D}^{\text{valid}}$, and test $\mathcal{D}^{\text{test}}$
  [e.g., 6 tournaments in $\mathcal{D}^{train}$, 2 tournaments in $\mathcal{D}^{valid}$, and 2 tournaments in $\mathcal{D}^{test}$]
for *each DP-model pipeline* **do**
  Tune hyperparameters by performing CV on $\mathcal{D}^{\text{train}}$
    [e.g., each hospital is its own fold in CV]
  Re-fit DP-model pipeline with optimal hyperaparameters on full $\mathcal{D}^{\text{train}}$

Example data splitting algorithm

Algorithm 1: Example using Repeated Data Splitting

Split data $\mathcal{D}$ into training $\mathcal{D}^{\text{train}}$, validation $\mathcal{D}^{\text{valid}}$, and test $\mathcal{D}^{\text{test}}$
  [e.g., 6 tournaments in $\mathcal{D}^{train}$, 2 tournaments in $\mathcal{D}^{valid}$, and 2 tournaments in $\mathcal{D}^{test}$]
for *each DP-model pipeline* **do**
  Tune hyperparameters by performing CV on $\mathcal{D}^{\text{train}}$
    [e.g., each hospital is its own fold in CV]
  Re-fit DP-model pipeline with optimal hyperaparameters on full $\mathcal{D}^{\text{train}}$
  Evaluate re-fitted DP-model pipeline on $\mathcal{D}^{\text{valid}}$
end
Select DP-model pipeline with best error on $\mathcal{D}^{\text{valid}}$

Example data splitting algorithm

Algorithm 1: Example using Repeated Data Splitting

Split data $\mathcal{D}$ into training $\mathcal{D}^{\text{train}}$, validation $\mathcal{D}^{\text{valid}}$, and test $\mathcal{D}^{\text{test}}$
  [e.g., 6 tournaments in $\mathcal{D}^{train}$, 2 tournaments in $\mathcal{D}^{valid}$, and 2 tournaments in $\mathcal{D}^{test}$]
for *each DP-model pipeline* **do**
  Tune hyperparameters by performing CV on $\mathcal{D}^{\text{train}}$
    [e.g., each hospital is its own fold in CV]
  Re-fit DP-model pipeline with optimal hyperaparameters on full $\mathcal{D}^{\text{train}}$
  Evaluate re-fitted DP-model pipeline on $\mathcal{D}^{\text{valid}}$
end
Select DP-model pipeline with best error on $\mathcal{D}^{\text{valid}}$
Re-fit DP-model pipeline on $(\mathcal{D}^{\text{train}}, \mathcal{D}^{\text{valid}})$
  [may include CV for tuning hyperparameters]
Evaluate re-fitted DP-model pipeline on $\mathcal{D}^{\text{test}} \rightarrow E_b$

Example data splitting algorithm

Algorithm 1: Example using Repeated Data Splitting

for $b = 1, \ldots, B$ **do**
  Split data $\mathcal{D}$ into training $\mathcal{D}^{\text{train}}$, validation $\mathcal{D}^{\text{valid}}$, and test $\mathcal{D}^{\text{test}}$
    [e.g., 6 tournaments in $\mathcal{D}^{train}$, 2 tournaments in $\mathcal{D}^{valid}$, and 2 tournaments in $\mathcal{D}^{test}$]
  for *each DP-model pipeline* **do**
    Tune hyperparameters by performing CV on $\mathcal{D}^{\text{train}}$
      [e.g., each hospital is its own fold in CV]
    Re-fit DP-model pipeline with optimal hyperaparameters on full $\mathcal{D}^{\text{train}}$
    Evaluate re-fitted DP-model pipeline on $\mathcal{D}^{\text{valid}}$
  end
  Select DP-model pipeline with best error on $\mathcal{D}^{\text{valid}}$
  Re-fit DP-model pipeline on $(\mathcal{D}^{\text{train}}, \mathcal{D}^{\text{valid}})$
    [may include CV for tuning hyperparameters]
  Evaluate re-fitted DP-model pipeline on $\mathcal{D}^{\text{test}} \rightarrow E_b$
end

Example data splitting algorithm

Algorithm 1: Example using Repeated Data Splitting

for $b = 1, \ldots, B$ **do**
  Split data $\mathcal{D}$ into training $\mathcal{D}^{\text{train}}$, validation $\mathcal{D}^{\text{valid}}$, and test $\mathcal{D}^{\text{test}}$
    [e.g., 6 tournaments in $\mathcal{D}^{train}$, 2 tournaments in $\mathcal{D}^{valid}$, and 2 tournaments in $\mathcal{D}^{test}$]
  for *each DP-model pipeline* **do**
    Tune hyperparameters by performing CV on $\mathcal{D}^{\text{train}}$
      [e.g., each hospital is its own fold in CV]
    Re-fit DP-model pipeline with optimal hyperaparameters on full $\mathcal{D}^{\text{train}}$
    Evaluate re-fitted DP-model pipeline on $\mathcal{D}^{\text{valid}}$
  end
  Select DP-model pipeline with best error on $\mathcal{D}^{\text{valid}}$
  Re-fit DP-model pipeline on $(\mathcal{D}^{\text{train}}, \mathcal{D}^{\text{valid}})$
    [may include CV for tuning hyperparameters]
  Evaluate re-fitted DP-model pipeline on $\mathcal{D}^{\text{test}} \rightarrow E_b$
end
Compute:

$$\text{Mean generalization error} = \frac{1}{B}\sum_{b=1}^{B} E_b \qquad \text{Variance of generalization error} = \frac{1}{B-1}\sum_{b=1}^{B}(E_b - \bar{E}_b)^2$$

Example data splitting algorithm

Algorithm 1: Example using Repeated Data Splitting

for $b = 1, \ldots, B$ **do**
  Split data $\mathcal{D}$ into training $\mathcal{D}^{\text{train}}$, validation $\mathcal{D}^{\text{valid}}$, and test $\mathcal{D}^{\text{test}}$
    [e.g., 6 tournaments in $\mathcal{D}^{train}$, 2 tournaments in $\mathcal{D}^{valid}$, and 2 tournaments in $\mathcal{D}^{test}$]
  for *each DP-model pipeline* **do**
    Tune hyperparameters by performing CV on $\mathcal{D}^{\text{train}}$
      [e.g., each hospital is its own fold in CV]
    Re-fit DP-model pipeline with optimal hyperaparameters on full $\mathcal{D}^{\text{train}}$
    Evaluate re-fitted DP-model pipeline on $\mathcal{D}^{\text{valid}}$
  end
  Select DP-model pipeline with best error on $\mathcal{D}^{\text{valid}}$
  Re-fit DP-model pipeline on $(\mathcal{D}^{\text{train}}, \mathcal{D}^{\text{valid}})$
    [may include CV for tuning hyperparameters]
  Evaluate re-fitted DP-model pipeline on $\mathcal{D}^{\text{test}} \rightarrow E_b$
end
Compute:

$$\text{Mean generalization error} = \frac{1}{B}\sum_{b=1}^{B} E_b \qquad \text{Variance of generalization error} = \frac{1}{B-1}\sum_{b=1}^{B}(E_b - \bar{E}_b)^2$$

If we want a "final" model for deployment, refit best DP-model pipeline on full $\mathcal{D}$
  [again, may include CV for tuning hyperparameters]

Example data splitting algorithm

Algorithm 1: Example using Repeated Data Splitting

for $b = 1, \ldots, B$ **do** *

  Split data $\mathcal{D}$ into training $\mathcal{D}^{\text{train}}$, validation $\mathcal{D}^{\text{valid}}$, and test $\mathcal{D}^{\text{test}}$

    [e.g., 6 tournaments in $\mathcal{D}^{train}$, 2 tournaments in $\mathcal{D}^{valid}$, and 2 tournaments in $\mathcal{D}^{test}$]

  for *each DP-model pipeline* **do**

    Tune hyperparameters by performing CV on $\mathcal{D}^{\text{train}}$

      [e.g., each hospital is its own fold in CV]

    Re-fit DP-model pipeline with optimal hyperaparameters on full $\mathcal{D}^{\text{train}}$

    Evaluate re-fitted DP-model pipeline on $\mathcal{D}^{\text{valid}}$

  end

  Select DP-model pipeline with best error on $\mathcal{D}^{\text{valid}}$

  Re-fit DP-model pipeline on $(\mathcal{D}^{\text{train}}, \mathcal{D}^{\text{valid}})$

    [may include CV for tuning hyperparameters]

  Evaluate re-fitted DP-model pipeline on $\mathcal{D}^{\text{test}} \rightarrow E_b$

end

Compute:

$$\text{Mean generalization error} = \frac{1}{B}\sum_{b=1}^{B} E_b \qquad \text{Variance of generalization error} = \frac{1}{B-1}\sum_{b=1}^{B}(E_b - \bar{E}_b)^2$$

If we want a "final" model for deployment, refit best DP-model pipeline on full $\mathcal{D}$

  [again, may include CV for tuning hyperparameters]

* Can replace repeated data splitting with CV

Key Takeaways of Data Splitting

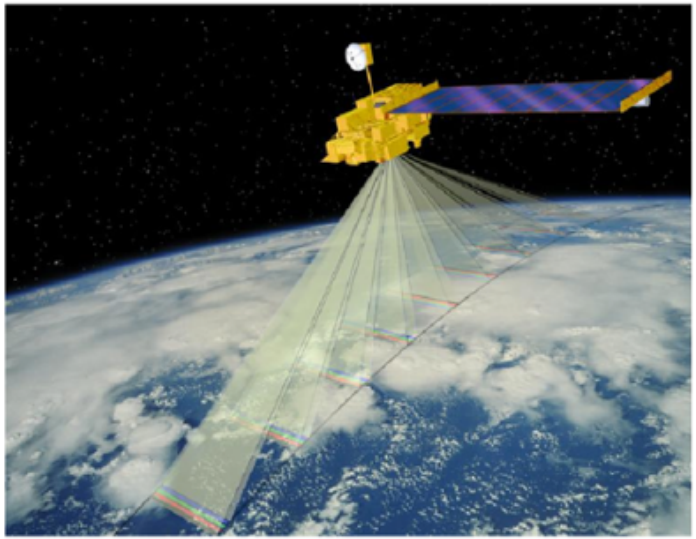
Data splitting is the key to assessing generalizability

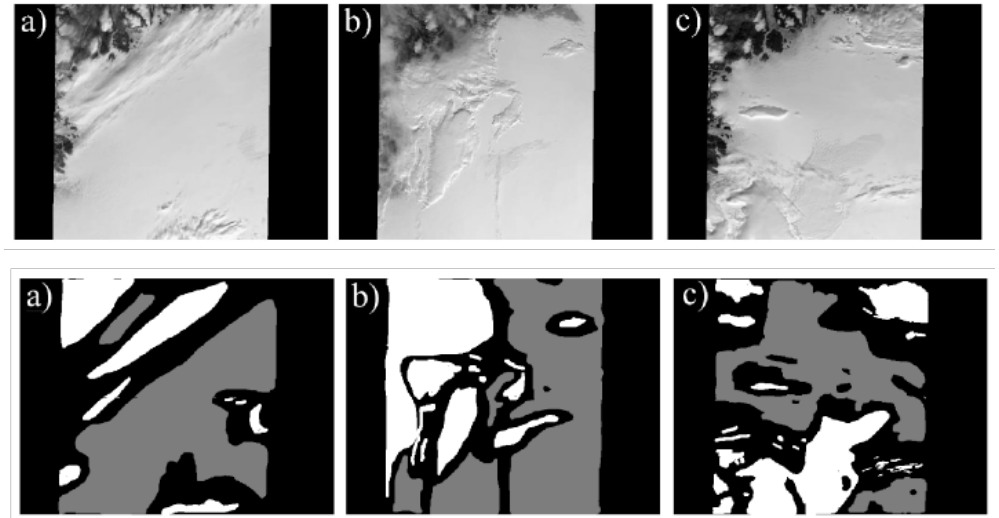
+ Need to decide how to (i) allocate samples into splits and (ii) splitting scheme
  (i) Should mimic the process of obtaining new data in the future
  (ii) Use training-validation-test split if selecting among multiple models;
  Use training-test if considering only one type of model
+ **Purpose of test set:** to obtain an unbiased estimate of the prediction error for your statistical learning **pipeline** on future data
  - Pipeline encompasses more than just the final prediction model. Includes the *process* of building your final prediction model
+ **IMPORTANT:** Do NOT touch the test set until the very end when you are ready to *evaluate* the generalization performance of your finalized pipeline; should only use the test set *once* to evaluate final pipeline

Introduction to Lab 3

Lab 3: Remote sensing for cloud detection

Goal: predict whether each pixel is a cloud or ice (from a glacier) on completely new satellite images





Original study: Shi et al. Daytime Arctic Cloud Detection Based on Multi-Angle Satellite Data with Case Studies

Lab 3 Notes

- Due March 6 at **11:59pm**
- “Free” 3-day late policy can be used to turn in this lab on the Monday after Spring Break (but please email me if you plan to do this)
- You are allowed to save your model results to a file/disk and read them in
  - Code to fit the models should be provided (e.g., set "#| eval: false" for that code chunk)
  - If you do this, upload your saved results to GitHub (so that I can also read them in)

In-Class Activity

In Lab 3, you are asked to choose between (at least) 2 models, where (at least) one requires hyperparameter tuning. You are also asked to report how well you think your best model would perform on new images.

Work in groups to:

- Make a data splitting plan for your cloud modeling pipeline
- How do you plan to tune hyperparameters in your model? How does this fit into your data splitting plan?
- How do you plan to implement your data splitting plan?
  - Python: check out sklearn data splitting guide
  - R: check out caret data splitting guide