Generalizability and Data Splitting

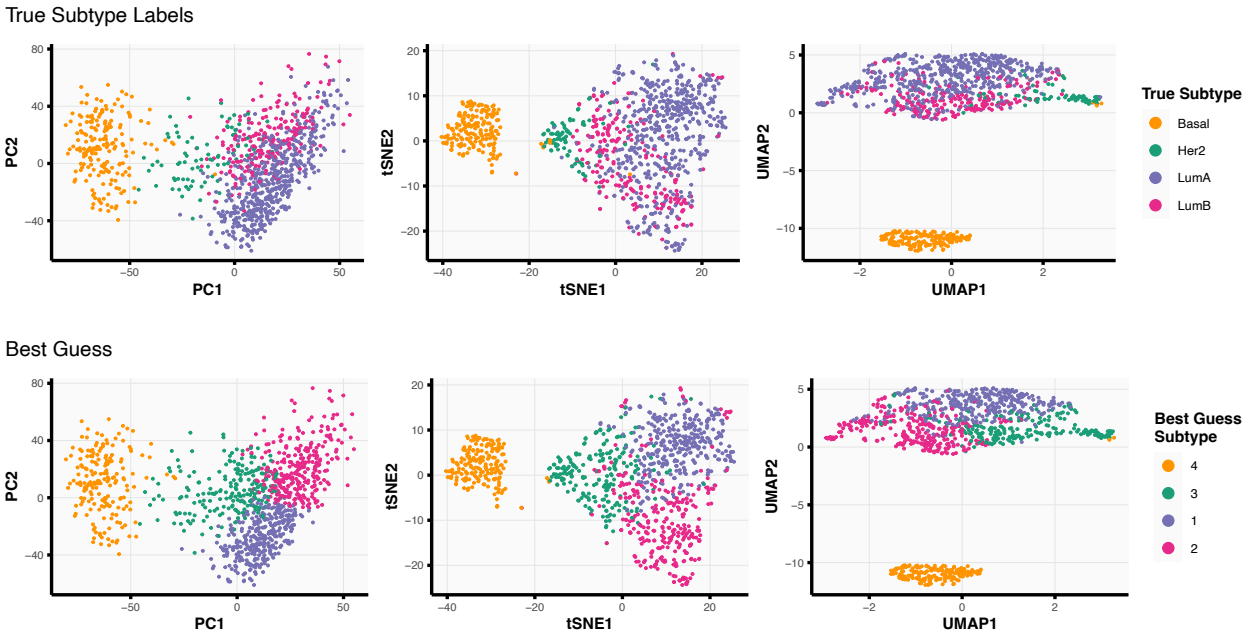
February 24, 2026

Mid-Semester Feedback Survey

Please fill out anonymous mid-semester feedback survey:
https://forms.gle/KoXjt3CbY91GKk7V8

Lab 2 Recap

- Lab 2 solutions have been released along with true breast cancer subtypes
  - See your dsip/lab2/results/true_clusters_plot.pdf for a quick visualization



[ChatGPT explanation](#) of different subtypes

Today's plan: Introduction to Supervised Learning

1. **Review** of last time
2. **Evaluation metrics** to assess generalizability
3. **Data splitting**
3. **Introduction to Lab 3**

Last Time: **The #1 Goal** of Supervised Learning

The goal of supervised learning is to learn insights from the current data that are generalizable to future, unseen data

Generalizability is closely related to:

- **Bias of the model:** on average, how wrong is the model compared to the truth?
- **Variance of the model:** if I retrained my model on a similar but new dataset, how much does my model change?

$$\underbrace{\mathbb{E}[(y - \hat{f}(x))^2]}_{\text{population MSE}} = \left(\text{Bias}(\hat{f}(x))\right)^2 + \text{Var}(\hat{f}(x)) + \underbrace{\sigma^2}_{\substack{\text{irreproducible} \\ \text{error}}}$$

(generalization error)

Regression and Classification Methods

	Linear/ Logistic	Ridge	LASSO	Elastic Net	Decision Trees	Random Forests	Neural Networks
Advantages	• Simple • Can do inference (with the usual assumptions)	• Good with correlated features • MSE Existence Theorem	• Feature selection and sparsity • Interpretable	• Compromise between ridge and LASSO	• Interpretable • Non-linear	• Non-linear • Powerful predictors	• Can naturally model very complex data structures • Powerful predictors
Disadvantages	• Can be unstable • Major problems when p > n	• Dense model is not interpretable • No automatic feature selection	• Often selects 1 feature from correlated group • Gives lower accuracy if truth is not sparse	• More difficult to tune two parameters in practice	• Unstable • Greedy, axis-aligned splits	• Not as interpretable as decision trees • Not efficient at learning smooth structures	• Difficult to tune • Generally requires lots of data

and more... (e.g., K nearest neighbors, support vector machines, bagging, ensembling)

Today: Revisiting **the #1 Goal** of Supervised Learning

The goal of supervised learning is to learn insights from the current data that are generalizable to future, unseen data

Q: How do we accurately **assess** generalizability (i.e., how well our model will do on new data)?

+ Prediction performance metrics
+ Data splitting

Prediction Performance Metrics

Regression Metrics

True (observed) responses: $\mathbf{y} = (y_1, \ldots, y_n)^\top \in \mathbb{R}^n$

Predicted responses: $\hat{\mathbf{y}} = (\hat{y}_1, \ldots, \hat{y}_n)^\top \in \mathbb{R}^n$

Metric	Definition	When to use

Regression Metrics

True (observed) responses: $\mathbf{y} = (y_1, \ldots, y_n)^\top \in \mathbb{R}^n$

Predicted responses: $\hat{\mathbf{y}} = (\hat{y}_1, \ldots, \hat{y}_n)^\top \in \mathbb{R}^n$

Metric	Definition	When to use
Mean squared error (MSE)	$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$	- Most popular and arguably the default metric for regression

Regression Metrics

True (observed) responses: $\mathbf{y} = (y_1, \ldots, y_n)^\top \in \mathbb{R}^n$

Predicted responses: $\hat{\mathbf{y}} = (\hat{y}_1, \ldots, \hat{y}_n)^\top \in \mathbb{R}^n$

Metric	Definition	When to use
Mean squared error (MSE)	$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$	- Most popular and arguably the default metric for regression
Root mean squared error (RMSE)	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$	- Most popular and arguably the default metric for regression - Error measured on same scale/units as y

Regression Metrics

True (observed) responses: $\mathbf{y} = (y_1, \ldots, y_n)^\top \in \mathbb{R}^n$

Predicted responses: $\hat{\mathbf{y}} = (\hat{y}_1, \ldots, \hat{y}_n)^\top \in \mathbb{R}^n$

Metric	Definition	When to use
Mean squared error (MSE)	$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$	- Most popular and arguably the default metric for regression
Root mean squared error (RMSE)	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$	- Most popular and arguably the default metric for regression - Error measured on same scale/units as y
R^2	$1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} = 1 - \frac{n}{n-1}\frac{MSE(\mathbf{y}, \hat{\mathbf{y}})}{\hat{Var}(\mathbf{y})}$	- Normalized MSE (normalized by variance of observed responses)

Regression Metrics

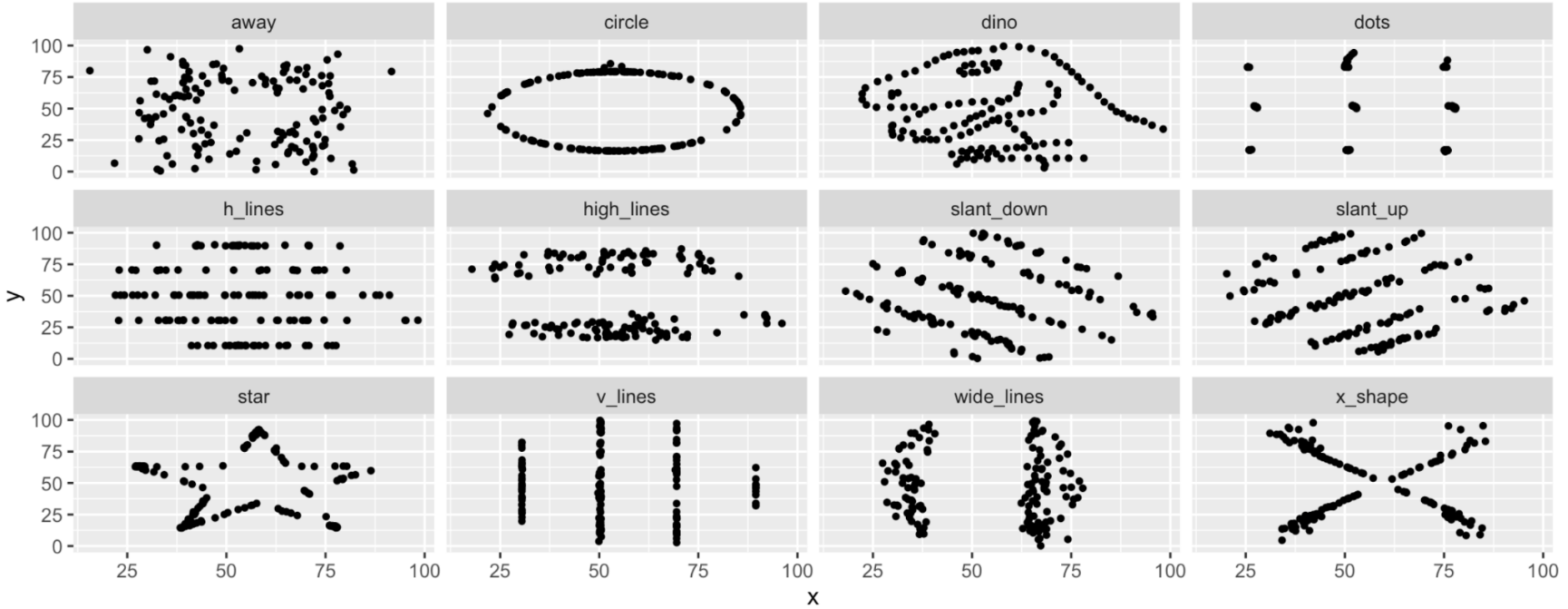
True (observed) responses: $\mathbf{y} = (y_1, \ldots, y_n)^\top \in \mathbb{R}^n$

Predicted responses: $\hat{\mathbf{y}} = (\hat{y}_1, \ldots, \hat{y}_n)^\top \in \mathbb{R}^n$

Metric	Definition	When to use
Mean squared error (MSE)	$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$	- Most popular and arguably the default metric for regression
Root mean squared error (RMSE)	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$	- Most popular and arguably the default metric for regression - Error measured on same scale/units as y
R^2	$1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} = 1 - \frac{n}{n-1}\frac{MSE(\mathbf{y}, \hat{\mathbf{y}})}{\hat{Var}(\mathbf{y})}$	- Normalized MSE (normalized by variance of observed responses)
Mean absolute error (MAE)	$\frac{1}{n}\sum_{i=1}^{n}\lvert y_i - \hat{y}_i\rvert$	- More robust to outliers than MSE/RMSE

We need to go beyond a single summary statistic

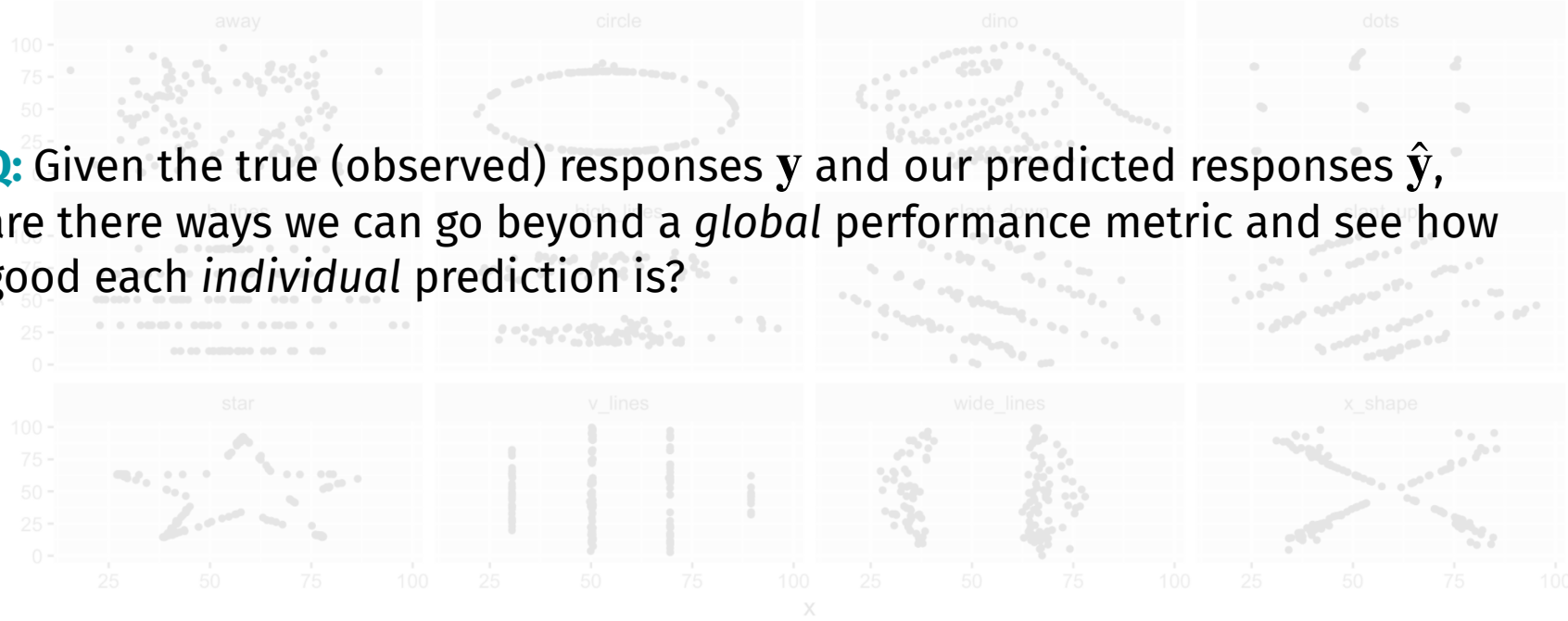
Each of these has the same mean, standard deviation, variance, and correlation



"The Datasaurus Dozen" [Matejka and Fitzmaurice (2017)]

We need to go beyond a single summary statistic

Each of these has the same mean, standard deviation, variance, and correlation



Q: Given the true (observed) responses $\mathbf{y}$ and our predicted responses $\hat{\mathbf{y}}$, are there ways we can go beyond a *global* performance metric and see how good each *individual* prediction is?

"The Datasaurus Dozen" [Matejka and Fitzmaurice (2017)]

Going beyond "global" summary metrics...

Plot true (observed) responses $\mathbf{y}$ versus predicted responses $\hat{\mathbf{y}}$

- Allows for further investigation into "odd" observations

Output of classification methods

True (observed) class labels

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Predicted class labels

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_n \end{bmatrix}$$

$$\hat{y}_i \in \{c_1, \ldots, c_K\}$$

Output of classification methods

True (observed) class labels

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Predicted class labels

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_n \end{bmatrix}$$

Predicted class probabilities

$$\hat{\mathbf{p}} = \begin{bmatrix} Pr(\hat{y}_1 = c_1) & \dots & Pr(\hat{y}_1 = c_K) \\ \vdots & \ddots & \vdots \\ Pr(\hat{y}_n = c_1) & \dots & Pr(\hat{y}_n = c_K) \end{bmatrix}$$

$$\hat{y}_i \in \{c_1, \dots, c_K\}$$

Output of classification methods

True (observed) class labels

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Predicted class labels

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_n \end{bmatrix}$$

$$\hat{y}_i \in \{c_1, \ldots, c_K\}$$

Predicted class probabilities

$$\hat{\mathbf{p}} = \begin{bmatrix} Pr(\hat{y}_1 = c_1) & \ldots & Pr(\hat{y}_1 = c_K) \\ \vdots & \ddots & \vdots \\ Pr(\hat{y}_n = c_1) & \ldots & Pr(\hat{y}_n = c_K) \end{bmatrix}$$

↑ Probability of being in class 1

Output of classification methods

True (observed) class labels

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Predicted class labels

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_n \end{bmatrix}$$

$$\hat{y}_i \in \{c_1, \ldots, c_K\}$$

Predicted class probabilities

$$\hat{\mathbf{p}} = \begin{bmatrix} Pr(\hat{y}_1 = c_1) & \ldots & Pr(\hat{y}_1 = c_K) \\ \vdots & \ddots & \vdots \\ Pr(\hat{y}_n = c_1) & \ldots & Pr(\hat{y}_n = c_K) \end{bmatrix}$$

Probability of being in class 1

Probability of being in class K

Output of classification methods

True (observed) class labels

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Predicted class labels

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_n \end{bmatrix}$$

$$\hat{y}_i \in \{c_1, \ldots, c_K\}$$

Predicted class probabilities

$$\hat{\mathbf{p}} = \begin{bmatrix} Pr(\hat{y}_1 = c_1) & \ldots & Pr(\hat{y}_1 = c_K) \\ \vdots & \ddots & \vdots \\ Pr(\hat{y}_n = c_1) & \ldots & Pr(\hat{y}_n = c_K) \end{bmatrix}$$

Probability of being in class 1

Probability of being in class K

$$\hat{\mathbf{p}} = \begin{bmatrix} Pr(\hat{y}_1 = 1) \\ \vdots \\ Pr(\hat{y}_n = 1) \end{bmatrix}$$

For binary classification, this is typically rewritten as a column vector (not an n x 2 matrix)

Output of classification methods

True (observed) class labels

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Predicted class labels

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_n \end{bmatrix}$$

Predicted class probabilities

$$\hat{\mathbf{p}} = \begin{bmatrix} Pr(\hat{y}_1 = c_1) & \dots & Pr(\hat{y}_1 = c_K) \\ \vdots & \ddots & \vdots \\ Pr(\hat{y}_n = c_1) & \dots & Pr(\hat{y}_n = c_K) \end{bmatrix}$$

↑ Probability of being in class 1

↑ Probability of being in class K

$$\hat{y}_i \in \{c_1, \dots, c_K\}$$

$$\hat{\mathbf{p}} = \begin{bmatrix} Pr(\hat{y}_1 = 1) \\ \vdots \\ Pr(\hat{y}_n = 1) \end{bmatrix}$$

For binary classification, this is typically rewritten as a column vector (not an n x 2 matrix)

Classification Accuracy

- **Classification accuracy** $= \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\{y_i = \hat{y}_i\}$

$$= \frac{\#\text{ of correctly predicted class labels}}{\text{total }\#\text{ of samples}}$$

Classification Accuracy

- **Classification accuracy** $= \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\{y_i = \hat{y}_i\}$

$$= \frac{\#\text{ of correctly predicted class labels}}{\text{total }\#\text{ of samples}}$$

- Higher is better

Classification Accuracy

- **Classification accuracy** $= \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{y_i = \hat{y}_i\}$

$$= \frac{\#\text{ of correctly predicted class labels}}{\text{total }\#\text{ of samples}}$$

- Higher is better
- What if we have class imbalance?

Classification Accuracy

- **Classification accuracy** $= \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{y_i = \hat{y}_i\}$

$$= \frac{\#\text{ of correctly predicted class labels}}{\text{total }\#\text{ of samples}}$$

- Higher is better
- What if we have class imbalance?
  - Ex. Suppose we have a sample of 100 people and only 10 have the disease. If we always predict healthy, we get 90% classification accuracy!

Confusion Matrix

We can go beyond classification accuracy and break the predicted responses down into a **confusion matrix**

		Predicted condition	
	Total population $= P + N$	**Predicted positive (PP)**	**Predicted negative (PN)**
Actual condition	**Positive (P)** [a]	**True positive** (TP), hit[b]	**False negative** (FN), miss, underestimation
	Negative (N)[d]	**False positive** (FP), false alarm, overestimation	**True negative** (TN), correct rejection[e]

Source: Wikipedia

Other classification metrics based on **predicted classes**

Sources: [16][17][18][19][20][21][22][23] view · talk · edit

		Predicted condition			
	Total population $= P + N$	**Predicted positive (PP)**	**Predicted negative (PN)**	Informedness, bookmaker informedness (BM) $= TPR + TNR - 1$	Prevalence threshold (PT) $= \frac{\sqrt{TPR \times FPR} - FPR}{TPR - FPR}$
Actual condition	**Positive (P)** [a]	**True positive** (TP), hit[b]	**False negative** (FN), miss, underestimation	True positive rate (TPR), recall, sensitivity (SEN), probability of detection, hit rate, power $= \frac{TP}{P} = 1 - FNR$	False negative rate (FNR), miss rate type II error [c] $= \frac{FN}{P} = 1 - TPR$
	Negative (N)[d]	**False positive** (FP), false alarm, overestimation	**True negative** (TN), correct rejection[e]	False positive rate (FPR), probability of false alarm, fall-out type I error [f] $= \frac{FP}{N} = 1 - TNR$	True negative rate (TNR), specificity (SPC), selectivity $= \frac{TN}{N} = 1 - FPR$
	Prevalence $= \frac{P}{P + N}$	Positive predictive value (PPV), precision $= \frac{TP}{PP} = 1 - FDR$	False omission rate (FOR) $= \frac{FN}{PN} = 1 - NPV$	Positive likelihood ratio (LR+) $= \frac{TPR}{FPR}$	Negative likelihood ratio (LR−) $= \frac{FNR}{TNR}$
	Accuracy (ACC) $= \frac{TP + TN}{P + N}$	False discovery rate (FDR) $= \frac{FP}{PP} = 1 - PPV$	Negative predictive value (NPV) $= \frac{TN}{PN} = 1 - FOR$	Markedness (MK), deltaP (Δp) $= PPV + NPV - 1$	Diagnostic odds ratio (DOR) $= \frac{LR+}{LR-}$
	Balanced accuracy (BA) $= \frac{TPR + TNR}{2}$	F_1 score $= \frac{2\,PPV \times TPR}{PPV + TPR} = \frac{2\,TP}{2\,TP + FP + FN}$	Fowlkes–Mallows index (FM) $= \sqrt{PPV \times TPR}$	Matthews correlation coefficient (MCC) $= \sqrt{TPR \times TNR \times PPV \times NPV} - \sqrt{FNR \times FPR \times FOR \times FDR}$	Threat score (TS), critical success index (CSI), Jaccard index $= \frac{TP}{TP + FN + FP}$

Source: Wikipedia

Other classification metrics based on **predicted classes**

Sources: [16][17][18][19][20][21][22][23] view · talk · edit

		Predicted condition			
	Total population $= P + N$	**Predicted positive (PP)**	**Predicted negative (PN)**	Informedness, bookmaker informedness (BM) $= TPR + TNR - 1$	Prevalence threshold (PT) $= \frac{\sqrt{TPR \times FPR} - FPR}{TPR - FPR}$
Actual condition	**Positive (P)** [a]	**True positive** (TP), hit[b]	**False negative** (FN), miss, underestimation	True positive rate (TPR), recall, sensitivity (SEN), probability of detection, hit rate, power $= \frac{TP}{P} = 1 - FNR$	False negative rate (FNR), miss rate type II error [c] $= \frac{FN}{P} = 1 - TPR$
	Negative (N)[d]	**False positive** (FP), false alarm, overestimation	**True negative** (TN), correct rejection[e]	False positive rate (FPR), probability of false alarm, fall-out type I error [f] $= \frac{FP}{N} = 1 - TNR$	True negative rate (TNR), specificity (SPC), selectivity $= \frac{TN}{N} = 1 - FPR$
	Prevalence $= \frac{P}{P + N}$	Positive predictive value (PPV), precision $= \frac{TP}{PP} = 1 - FDR$	False omission rate (FOR) $= \frac{FN}{PN} = 1 - NPV$	Positive likelihood ratio (LR+) $= \frac{TPR}{FPR}$	Negative likelihood ratio (LR−) $= \frac{FNR}{TNR}$
	Accuracy (ACC) $= \frac{TP + TN}{P + N}$	False discovery rate (FDR) $= \frac{FP}{PP} = 1 - PPV$	Negative predictive value (NPV) $= \frac{TN}{PN} = 1 - FOR$	Markedness (MK), deltaP (Δp) $= PPV + NPV - 1$	Diagnostic odds ratio (DOR) $= \frac{LR+}{LR-}$
	Balanced accuracy (BA) $= \frac{TPR + TNR}{2}$	F_1 score $= \frac{2\,PPV \times TPR}{PPV + TPR} = \frac{2\,TP}{2\,TP + FP + FN}$	Fowlkes–Mallows index (FM) $= \sqrt{PPV \times TPR}$	Matthews correlation coefficient (MCC) $= \sqrt{TPR \times TNR \times PPV \times NPV} - \sqrt{FNR \times FPR \times FOR \times FDR}$	Threat score (TS), critical success index (CSI), Jaccard index $= \frac{TP}{TP + FN + FP}$

Source: Wikipedia

Other classification metrics based on **predicted classes**

		Predicted condition		Sources: [16][17][18][19][20][21][22][23] view · talk · edit	
	Total population $= P + N$	**Predicted positive (PP)**	**Predicted negative (PN)**	Informedness, bookmaker informedness (BM) $= TPR + TNR - 1$	Prevalence threshold (PT) $= \frac{\sqrt{TPR \times FPR} - FPR}{TPR - FPR}$
Actual condition	**Positive (P)** [a]	**True positive** (TP), hit[b]	**False negative** (FN), miss, underestimation	True positive rate (TPR), recall, sensitivity (SEN), probability of detection, hit rate, power $= \frac{TP}{P} = 1 - FNR$	False negative rate (FNR), miss rate type II error [c] $= \frac{FN}{P} = 1 - TPR$
	Negative (N)[d]	**False positive** (FP), false alarm, overestimation	**True negative** (TN), correct rejection[e]	False positive rate (FPR), probability of false alarm, fall-out type I error [f] $= \frac{FP}{N} = 1 - TNR$	True negative rate (TNR), specificity (SPC), selectivity $= \frac{TN}{N} = 1 - FPR$
	Prevalence $= \frac{P}{P + N}$	Positive predictive value (PPV), precision $= \frac{TP}{PP} = 1 - FDR$	False omission rate (FOR) $= \frac{FN}{PN} = 1 - NPV$	Positive likelihood ratio (LR+) $= \frac{TPR}{FPR}$	Negative likelihood ratio (LR−) $= \frac{FNR}{TNR}$
	Accuracy (ACC) $= \frac{TP + TN}{P + N}$	False discovery rate (FDR) $= \frac{FP}{PP} = 1 - PPV$	Negative predictive value (NPV) $= \frac{TN}{PN} = 1 - FOR$	Markedness (MK), deltaP (Δp) $= PPV + NPV - 1$	Diagnostic odds ratio (DOR) $= \frac{LR+}{LR-}$
	Balanced accuracy (BA) $= \frac{TPR + TNR}{2}$	F_1 score $= \frac{2\,PPV \times TPR}{PPV + TPR} = \frac{2\,TP}{2\,TP + FP + FN}$	Fowlkes–Mallows index (FM) $= \sqrt{PPV \times TPR}$	Matthews correlation coefficient (MCC) $= \sqrt{TPR \times TNR \times PPV \times NPV} - \sqrt{FNR \times FPR \times FOR \times FDR}$	Threat score (TS), critical success index (CSI), Jaccard index $= \frac{TP}{TP + FN + FP}$

Source: Wikipedia

Other classification metrics based on **predicted classes**

		Predicted condition		Sources: [16][17][18][19][20][21][22][23] view · talk · edit	
	Total population $= P + N$	**Predicted positive (PP)**	**Predicted negative (PN)**	Informedness, bookmaker informedness (BM) $= TPR + TNR - 1$	Prevalence threshold (PT) $= \frac{\sqrt{TPR \times FPR} - FPR}{TPR - FPR}$
Actual condition	**Positive (P)** [a]	**True positive** (TP), hit[b]	**False negative** (FN), miss, underestimation	True positive rate (TPR), recall, sensitivity (SEN), probability of detection, hit rate, power $= \frac{TP}{P} = 1 - FNR$	False negative rate (FNR), miss rate type II error [c] $= \frac{FN}{P} = 1 - TPR$
	Negative (N)[d]	**False positive** (FP), false alarm, overestimation	**True negative** (TN), correct rejection[e]	False positive rate (FPR), probability of false alarm, fall-out type I error [f] $= \frac{FP}{N} = 1 - TNR$	True negative rate (TNR), specificity (SPC), selectivity $= \frac{TN}{N} = 1 - FPR$
	Prevalence $= \frac{P}{P + N}$	Positive predictive value (PPV), precision $= \frac{TP}{PP} = 1 - FDR$	False omission rate (FOR) $= \frac{FN}{PN} = 1 - NPV$	Positive likelihood ratio (LR+) $= \frac{TPR}{FPR}$	Negative likelihood ratio (LR−) $= \frac{FNR}{TNR}$
	Accuracy (ACC) $= \frac{TP + TN}{P + N}$	False discovery rate (FDR) $= \frac{FP}{PP} = 1 - PPV$	Negative predictive value (NPV) $= \frac{TN}{PN} = 1 - FOR$	Markedness (MK), deltaP (Δp) $= PPV + NPV - 1$	Diagnostic odds ratio (DOR) $= \frac{LR+}{LR-}$
	Balanced accuracy (BA) $= \frac{TPR + TNR}{2}$	F_1 score $= \frac{2\,PPV \times TPR}{PPV + TPR} = \frac{2\,TP}{2\,TP + FP + FN}$	Fowlkes–Mallows index (FM) $= \sqrt{PPV \times TPR}$	Matthews correlation coefficient (MCC) $= \sqrt{TPR \times TNR \times PPV \times NPV} - \sqrt{FNR \times FPR \times FOR \times FDR}$	Threat score (TS), critical success index (CSI), Jaccard index $= \frac{TP}{TP + FN + FP}$

Source: Wikipedia

Output of classification methods

True (observed) class labels

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Predicted class labels

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_n \end{bmatrix}$$

Predicted class probabilities

$$\hat{\mathbf{p}} = \begin{bmatrix} Pr(\hat{y}_1 = c_1) & \dots & Pr(\hat{y}_1 = c_K) \\ \vdots & \ddots & \vdots \\ Pr(\hat{y}_n = c_1) & \dots & Pr(\hat{y}_n = c_K) \end{bmatrix}$$

↑ Probability of being in class 1

↑ Probability of being in class K

$$\hat{y}_i \in \{c_1, \dots, c_K\}$$

$$\hat{\mathbf{p}} = \begin{bmatrix} Pr(\hat{y}_1 = 1) \\ \vdots \\ Pr(\hat{y}_n = 1) \end{bmatrix}$$

For binary classification, this is typically rewritten as a column vector (not an n x 2 matrix)

Output of classification methods

True (observed)
class labels

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Predicted
class labels

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_n \end{bmatrix}$$

$$\hat{y}_i \in \{c_1, \ldots, c_K\}$$

Predicted
class probabilities

$$\hat{\mathbf{p}} = \begin{bmatrix} Pr(\hat{y}_1 = c_1) & \ldots & Pr(\hat{y}_1 = c_K) \\ \vdots & \ddots & \vdots \\ Pr(\hat{y}_n = c_1) & \ldots & Pr(\hat{y}_n = c_K) \end{bmatrix}$$

Probability of being in class 1

Probability of being in class K

$$\hat{\mathbf{p}} = \begin{bmatrix} Pr(\hat{y}_1 = 1) \\ \vdots \\ Pr(\hat{y}_n = 1) \end{bmatrix}$$

For binary classification, this is typically rewritten as a column vector (not an n x 2 matrix)

Other classification metrics based on **predicted probabilities**

Other classification metrics based on **predicted probabilities**

- **Idea:** Different probability cutoffs/thresholds can be chosen to yield different predicted class labels

Other classification metrics based on **predicted probabilities**

- **Idea:** Different probability cutoffs/thresholds can be chosen to yield different predicted class labels
  - Ex. For a binary classification problem, if P(y = 1) > 0.5, we typically predict class 1 and class 0 otherwise. However, if the harm of a false positive is very high, we might want to choose a different (higher) cutoff

Other classification metrics based on **predicted probabilities**

- **Idea:** Different probability cutoffs/thresholds can be chosen to yield different predicted class labels
  - Ex. For a binary classification problem, if P(y = 1) > 0.5, we typically predict class 1 and class 0 otherwise. However, if the harm of a false positive is very high, we might want to choose a different (higher) cutoff
- This leads to metrics like

Other classification metrics based on **predicted probabilities**

- **Idea:** Different probability cutoffs/thresholds can be chosen to yield different predicted class labels
  - Ex. For a binary classification problem, if P(y = 1) > 0.5, we typically predict class 1 and class 0 otherwise. However, if the harm of a false positive is very high, we might want to choose a different (higher) cutoff
- This leads to metrics like
  - **Area under the ROC (AUROC)**, or

Other classification metrics based on **predicted probabilities**

- **Idea:** Different probability cutoffs/thresholds can be chosen to yield different predicted class labels
  - Ex. For a binary classification problem, if P(y = 1) > 0.5, we typically predict class 1 and class 0 otherwise. However, if the harm of a false positive is very high, we might want to choose a different (higher) cutoff
- This leads to metrics like
  - **Area under the ROC (AUROC)**, or
  - **Area under the precision-recall curve (AUPRC)**

Other classification metrics based on **predicted probabilities**

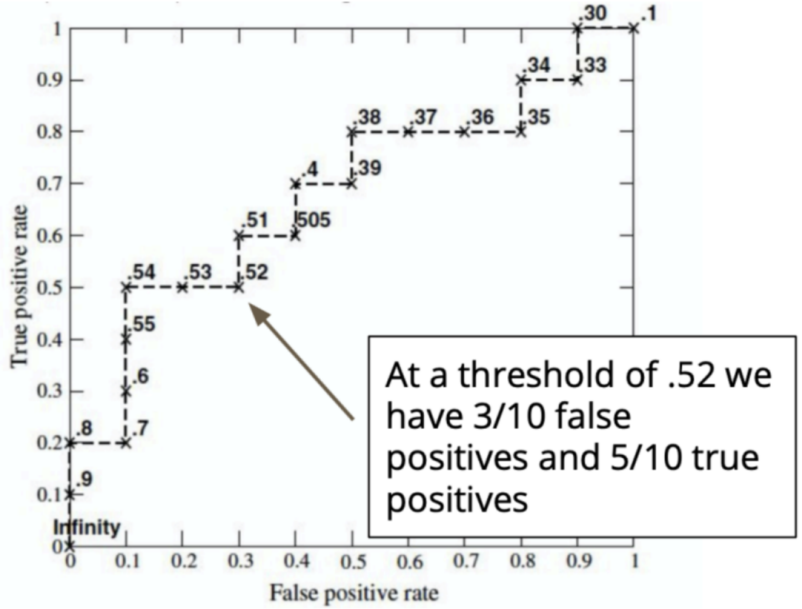
- **Idea:** Different probability cutoffs/thresholds can be chosen to yield different predicted class labels
  - Ex. For a binary classification problem, if P(y = 1) > 0.5, we typically predict class 1 and class 0 otherwise. However, if the harm of a false positive is very high, we might want to choose a different (higher) cutoff
- This leads to metrics like
  - **Area under the ROC (AUROC)**, or
  - **Area under the precision-recall curve (AUPRC)**

  which summarize the classification accuracy across many different choices of thresholds

Receiver operating characteristics (ROC) curve

- **ROC curve:** false positive rate (FPR) versus true positive rate (TPR)
  - FPR = (# false positives) / (# observed negative samples)
  - TPR = (# true positives) / (# observed positive samples)

Inst#	Class	Score	Inst#	Class	Score
1	p	.9	11	p	.4
2	p	.8	12	n	.39
3	n	.7	13	p	.38
4	p	.6	14	n	.37
5	p	.55	15	n	.36
6	p	.54	16	n	.35
7	n	.53	17	p	.34
8	n	.52	18	n	.33
9	p	.51	19	p	.30
10	n	.505	20	n	.1

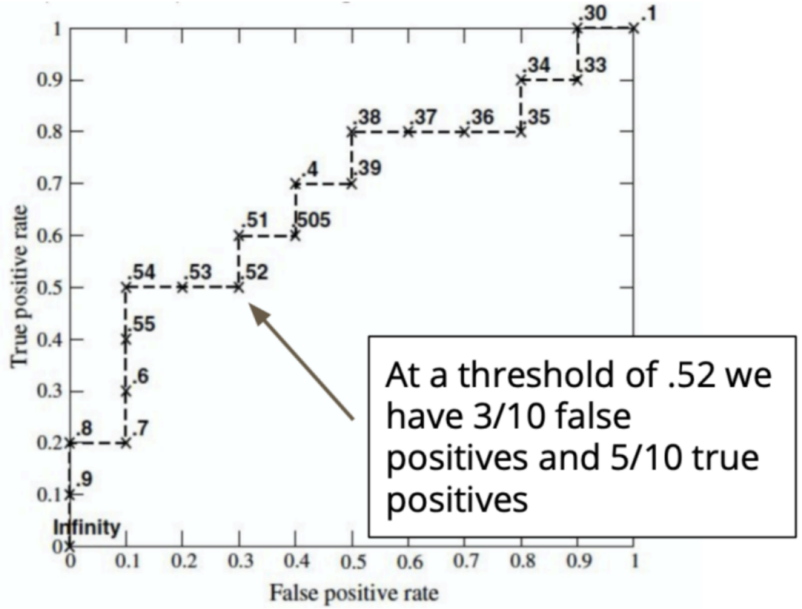


Source: Fawcett (2005)

Receiver operating characteristics (ROC) curve

- **ROC curve:** false positive rate (FPR) versus true positive rate (TPR)
  - FPR = (# false positives) / (# observed negative samples)
  - TPR = (# true positives) / (# observed positive samples)

Inst#	Class	Score	Inst#	Class	Score
1	p	.9	11	p	.4
2	p	.8	12	n	.39
3	n	.7	13	p	.38
4	p	.6	14	n	.37
5	p	.55	15	n	.36
6	p	.54	16	n	.35
7	n	.53	17	p	.34
8	n	.52	18	n	.33
9	p	.51	19	p	.30
10	n	.505	20	n	.1



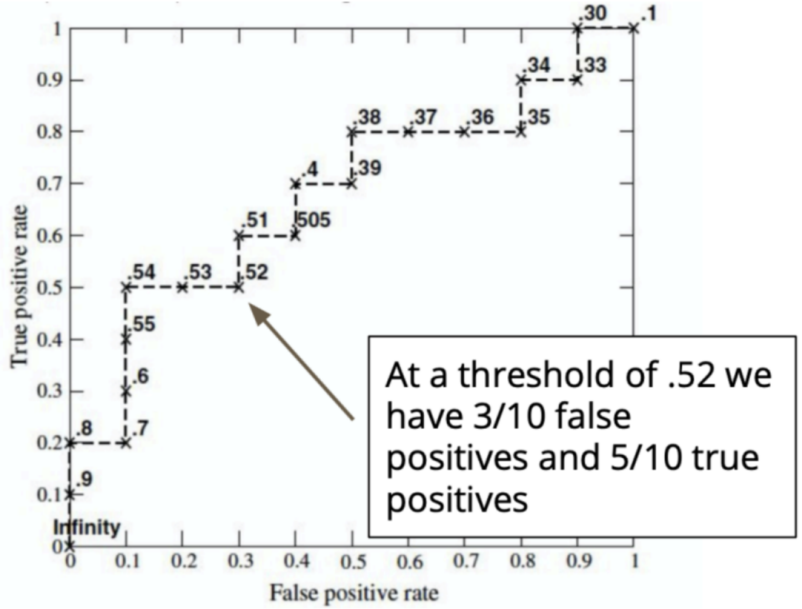
Source: Fawcett (2005)

Receiver operating characteristics (ROC) curve

- **ROC curve:** false positive rate (FPR) versus true positive rate (TPR)
  - FPR = (# false positives) / (# observed negative samples)
  - TPR = (# true positives) / (# observed positive samples)

Inst#	Class	Score	Inst#	Class	Score
1	p	.9	11	p	.4
2	p	.8	12	n	.39
3	n	.7	13	p	.38
4	p	.6	14	n	.37
5	p	.55	15	n	.36
6	p	.54	16	n	.35
7	n	.53	17	p	.34
8	n	.52	18	n	.33
9	p	.51	19	p	.30
10	n	.505	20	n	.1

Classify positive ↑



Source: Fawcett (2005)

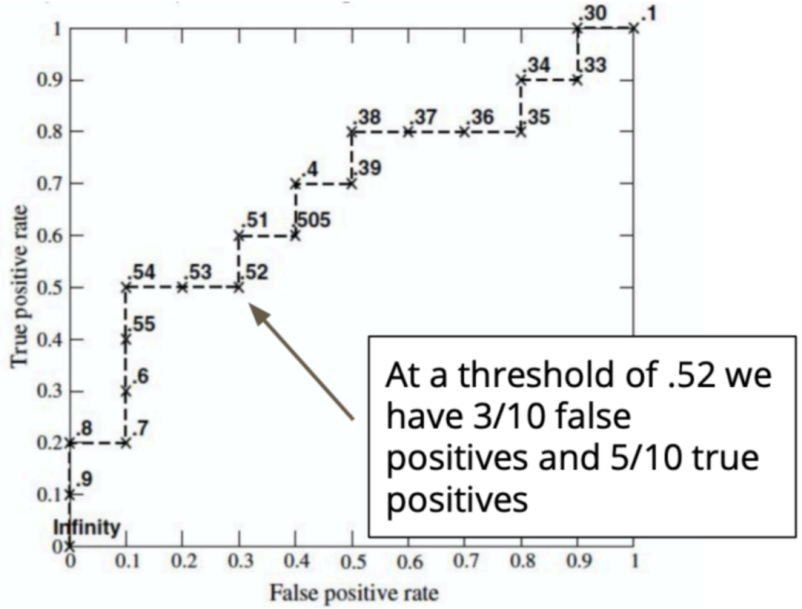
Receiver operating characteristics (ROC) curve

- **ROC curve:** false positive rate (FPR) versus true positive rate (TPR)
  - FPR = (# false positives) / (# observed negative samples)
  - TPR = (# true positives) / (# observed positive samples)

Inst#	Class	Score	Inst#	Class	Score
1	p	.9	11	p	.4
2	p	.8	12	n	.39
3	n	.7	13	p	.38
4	p	.6	14	n	.37
5	p	.55	15	n	.36
6	p	.54	16	n	.35
7	n	.53	17	p	.34
8	n	.52	18	n	.33
9	p	.51	19	p	.30
10	n	.505	20	n	.1

Classify positive

Classify negative



Source: Fawcett (2005)

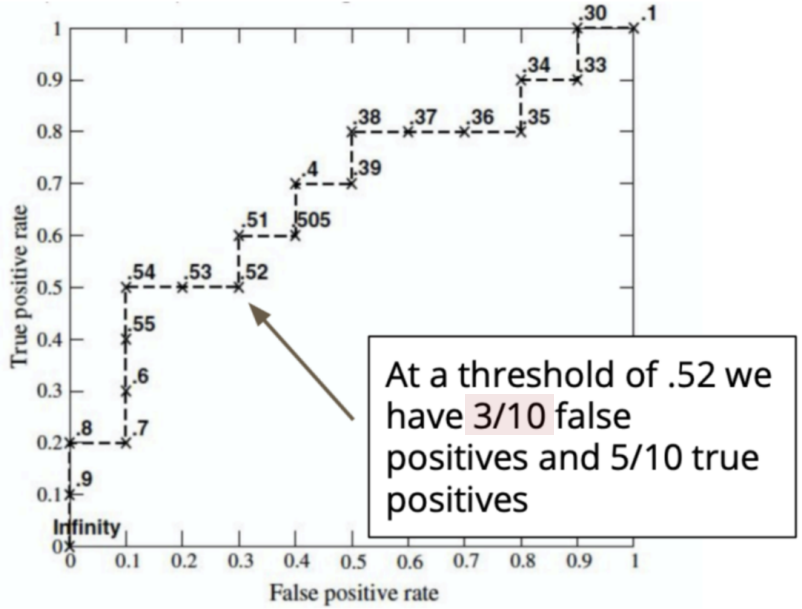
Receiver operating characteristics (ROC) curve

- **ROC curve:** false positive rate (FPR) versus true positive rate (TPR)
  - FPR = (# false positives) / (# observed negative samples)
  - TPR = (# true positives) / (# observed positive samples)

Inst#	Class	Score	Inst#	Class	Score
1	p	.9	11	p	.4
2	p	.8	12	n	.39
3	n	.7	13	p	.38
4	p	.6	14	n	.37
5	p	.55	15	n	.36
6	p	.54	16	n	.35
7	n	.53	17	p	.34
8	n	.52	18	n	.33
9	p	.51	19	p	.30
10	n	.505	20	n	.1

Classify positive

Classify negative



Source: Fawcett (2005)

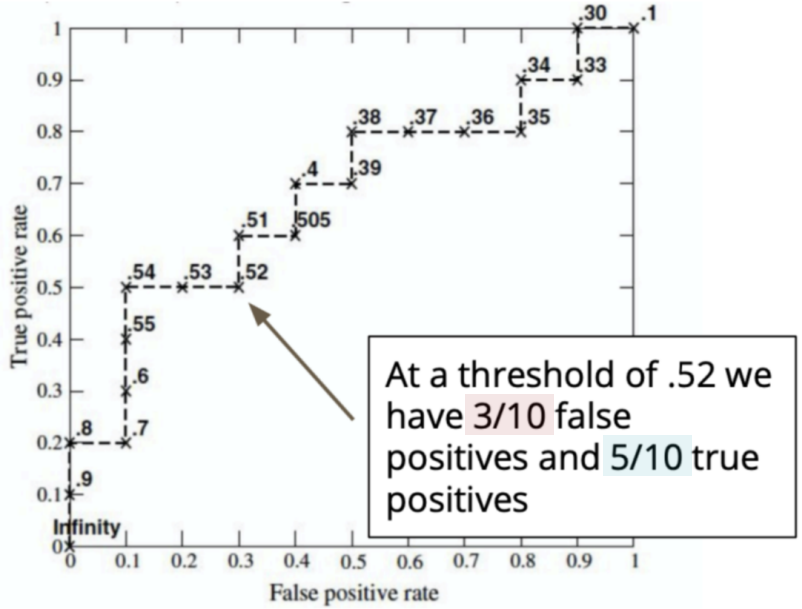
Receiver operating characteristics (ROC) curve

- **ROC curve:** false positive rate (FPR) versus true positive rate (TPR)
  - FPR = (# false positives) / (# observed negative samples)
  - TPR = (# true positives) / (# observed positive samples)

Inst#	Class	Score	Inst#	Class	Score
1	p	.9	11	p	.4
2	p	.8	12	n	.39
3	n	.7	13	p	.38
4	p	.6	14	n	.37
5	p	.55	15	n	.36
6	p	.54	16	n	.35
7	n	.53	17	p	.34
8	n	.52	18	n	.33
9	p	.51	19	p	.30
10	n	.505	20	n	.1

Classify positive

Classify negative

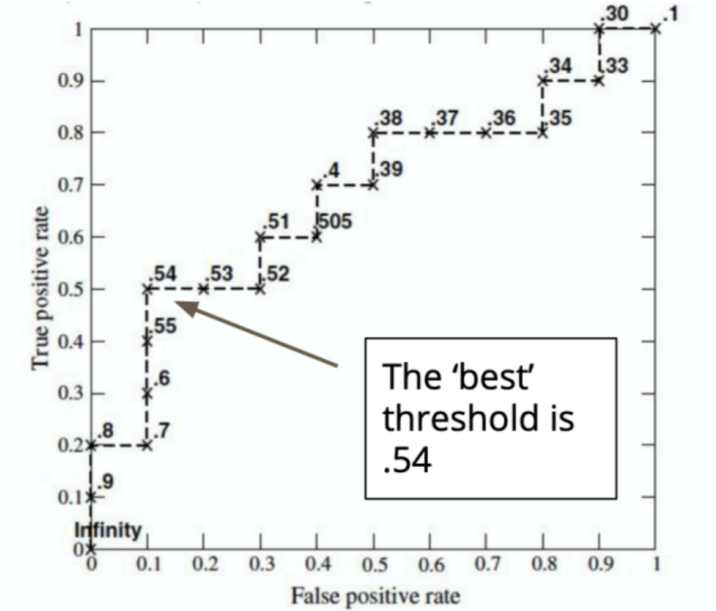


Source: Fawcett (2005)

Receiver operating characteristics (ROC) curve

- **ROC curve:** false positive rate (FPR) versus true positive rate (TPR)
  - FPR = (# false positives) / (# observed negative samples)
  - TPR = (# true positives) / (# observed positive samples)

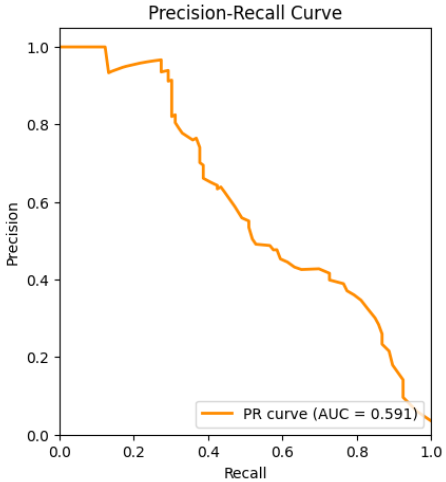
- The **area under the ROC curve (AUROC)** summarizes the results across all possible thresholds
- The **"best" threshold** is often chosen to be the one closest to the top left corner of this plot (Youden's index)
  - Choose point which has the smallest distance to (0, 1)



Source: Fawcett (2005)

Precision-recall curve

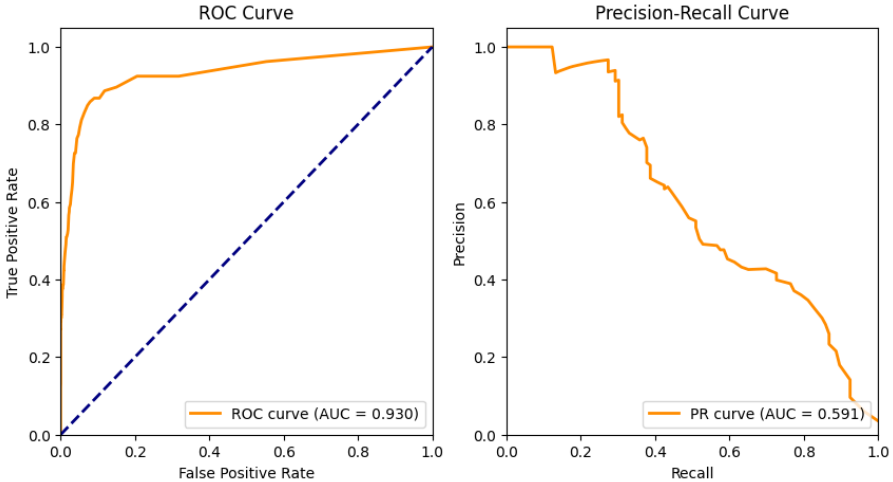
- **PR curve:** precision versus recall
  - Precision = (# true positives) / (# predicted positive samples)
  - Recall (aka TPR) = (# true positives) / (# observed positive samples)
- The **area under the PR curve (AUPRC)** summarizes the results across all possible thresholds



Comparing ROC and PR curves

- **PR curve:** precision versus recall
  - Precision = (# true positives) / (# predicted positive samples)
  - Recall (aka TPR) = (# true positives) / (# observed positive samples)
- **ROC curve:** false positive rate (FPR) versus true positive rate (TPR)
  - FPR = (# false positives) / (# observed negative samples)
  - TPR (aka recall) = (# true positives) / (# observed positive samples)

AUROC
baseline value = 0.5



AUPRC
baseline value =
proportion of
samples in
“positive” class

Sidebar: Comparing "Baseline" ROC and PR curves

"Baseline" = predictions are completely independent of the true observed responses

ROC

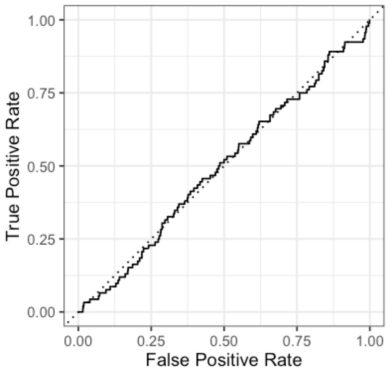
$$\text{Recall/TPR} = \frac{\#\text{ True Positives}}{\#\text{ Observed Positives}}$$

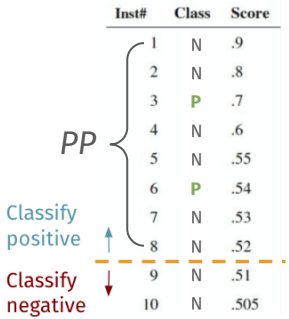
$$\text{FPR} = \frac{\#\text{ False Positives}}{\#\text{ Observed Negatives}}$$

AUROC Baseline = 0.5

$$TPR = \frac{\pi \cdot \text{PP}}{\pi \cdot n} = \frac{\text{PP}}{n}$$

$$FPR = \frac{(1-\pi) \cdot \text{PP}}{(1-\pi) \cdot n} = \frac{\text{PP}}{n}$$

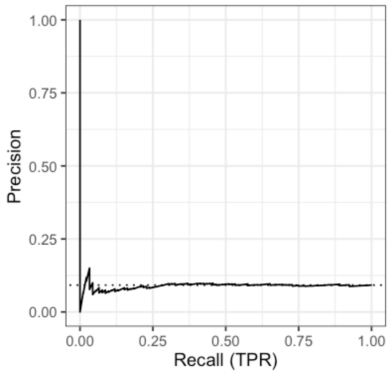




PR

$$\text{Recall/TPR} = \frac{\#\text{ True Positives}}{\#\text{ Observed Positives}}$$

$$\text{Precision} = \frac{\#\text{ True Positives}}{\#\text{ Predicted Positives}}$$



AUPRC Baseline = proportion of samples in positive class

$$TPR = \frac{\pi \cdot \text{PP}}{\pi \cdot n} = \frac{\text{PP}}{n}$$

$$\text{Precision} = \frac{\pi \cdot \text{PP}}{\text{PP}} = \pi$$

where # samples = n, proportion of positive class = π, and # predicted positives = PP

Comparing ROC and PR curves

- **PR curve:** precision versus recall
  - Precision = (# true positives) / (# predicted positive samples)
  - Recall (aka TPR) = (# true positives) / (# observed positive samples)
- **ROC curve:** false positive rate (FPR) versus true positive rate (TPR)
  - FPR = (# false positives) / (# observed negative samples)
  - TPR = (# true positives) / (# observed positive samples)

Comparing ROC and PR curves

- **PR curve:** precision versus recall
  - Precision = (# true positives) / (# predicted positive samples)
  - Recall (aka TPR) = (# true positives) / (# observed positive samples)
- **ROC curve:** false positive rate (FPR) versus true positive rate (TPR)
  - FPR = (# false positives) / (# observed negative samples)
  - TPR = (# true positives) / (# observed positive samples)

- For highly imbalance problems where # negative (N) >> # positive (P) samples,

Comparing ROC and PR curves

- **PR curve:** precision versus recall
  - Precision = (# true positives) / (# predicted positive samples)
  - Recall (aka TPR) = (# true positives) / (# observed positive samples)
- **ROC curve:** false positive rate (FPR) versus true positive rate (TPR)
  - FPR = (# false positives) / (# observed negative samples)
  - TPR = (# true positives) / (# observed positive samples)

- For highly imbalance problems where # negative (N) >> # positive (P) samples,
  - FPR will always look pretty good no matter what FP is since FPR is dividing by N (a big number)

Comparing ROC and PR curves

- **PR curve:** precision versus recall
  - Precision = (# true positives) / (# predicted positive samples)
  - Recall (aka TPR) = (# true positives) / (# observed positive samples)
- **ROC curve:** false positive rate (FPR) versus true positive rate (TPR)
  - FPR = (# false positives) / (# observed negative samples)
  - TPR = (# true positives) / (# observed positive samples)

- For highly imbalance problems where # negative (N) >> # positive (P) samples,
  - FPR will always look pretty good no matter what FP is since FPR is dividing by N (a big number)
  - Precision focuses instead on the predicted positive samples and avoids using N

Comparing ROC and PR curves

- **PR curve:** precision versus recall
  - Precision = (# true positives) / (# predicted positive samples)
  - Recall (aka TPR) = (# true positives) / (# observed positive samples)
- **ROC curve:** false positive rate (FPR) versus true positive rate (TPR)
  - FPR = (# false positives) / (# observed negative samples)
  - TPR = (# true positives) / (# observed positive samples)

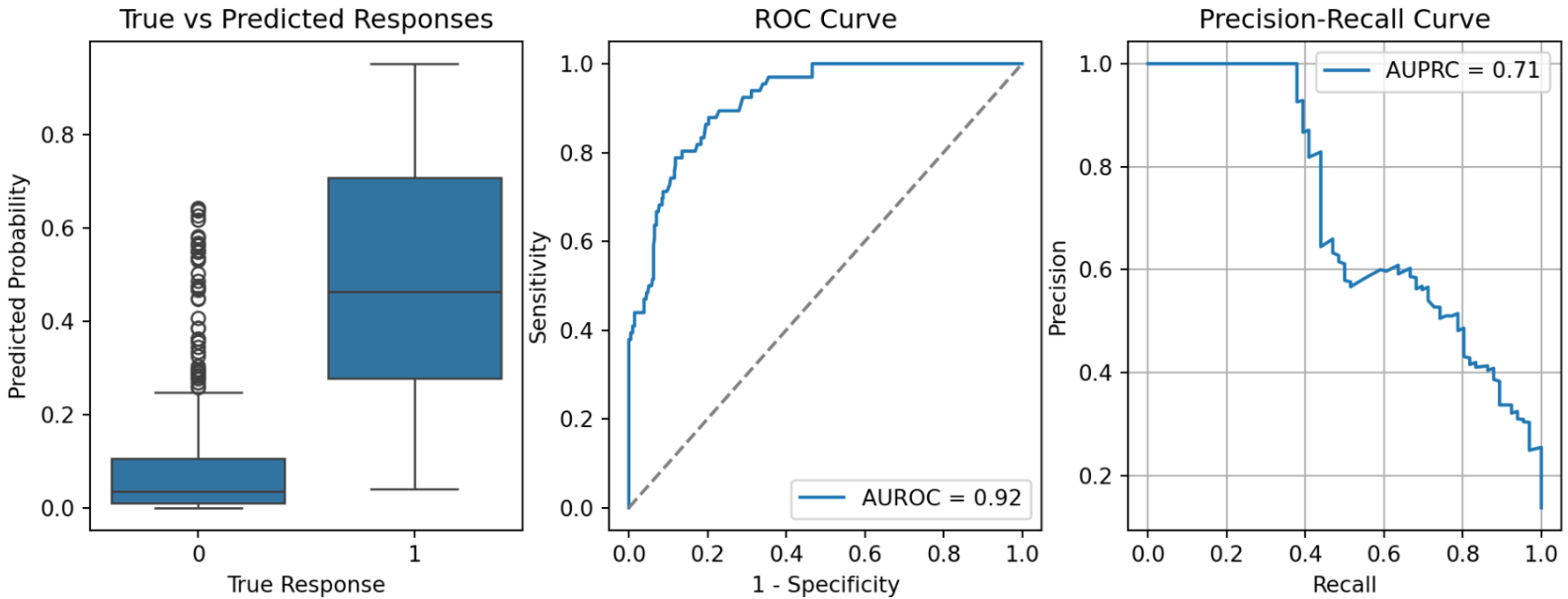
- For highly imbalance problems where # negative (N) >> # positive (P) samples,
  - FPR will always look pretty good no matter what FP is since FPR is dividing by N (a big number)
  - Precision focuses instead on the predicted positive samples and avoids using N
- As such, precision-recall curves are often preferred when there is high class imbalance

Comparing ROC and PR curves

- **PR curve:** precision versus recall
  - Precision = (# true positives) / (# predicted positive samples)
  - Recall (aka TPR) = (# true positives) / (# observed positive samples)
- **ROC curve:** false positive rate (FPR) versus true positive rate (TPR)
  - FPR = (# false positives) / (# observed negative samples)
  - TPR = (# true positives) / (# observed positive samples)

- For highly imbalance problems where # negative (N) >> # positive (P) samples,
  - FPR will always look pretty good no matter what FP is since FPR is dividing by N (a big number)
  - Precision focuses instead on the predicted positive samples and avoids using N
- As such, precision-recall curves are often preferred when there is high class imbalance
- **My general advice: always look at multiple metrics**

Going beyond global summary metrics...



Data Splitting

Motivating Thought Experiment

Ultimate question: How well does my model perform on future, unseen data?

Consider the following workflow:

1. **Train/fit** prediction model on data $\mathbf{X}$
2. **Evaluate** prediction model on same dataset $\mathbf{X} \rightarrow$ demonstrates Z% accuracy
3. **Generalizability:**
   "I expect my model to achieve Z% accuracy when deployed for future data"

Motivating Thought Experiment

Ultimate question: How well does my model perform on future, unseen data?

Consider the following workflow:

1. **Train/fit** prediction model on data $\mathbf{X}$
2. **Evaluate** prediction model on same dataset $\mathbf{X} \rightarrow$ demonstrates Z% accuracy
3. **Generalizability:**
   "I expect my model to achieve Z% accuracy when deployed for future data"

Problem: most likely over-fitting to the existing training data

Overview of Data Splitting

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

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:

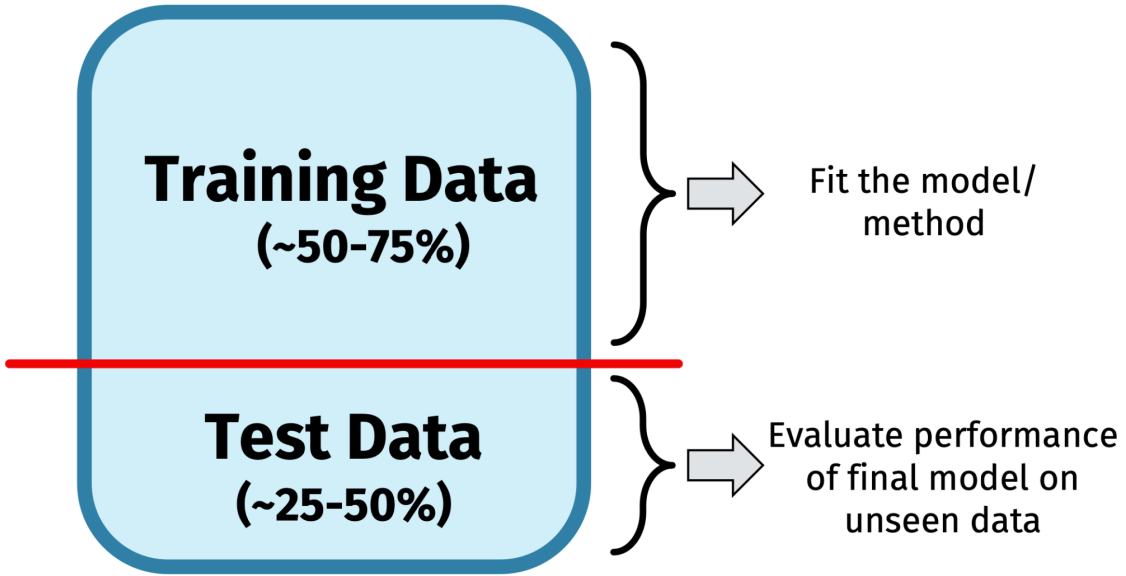
(ignoring choice of hyperparameters
and possibility of multiple models)

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)



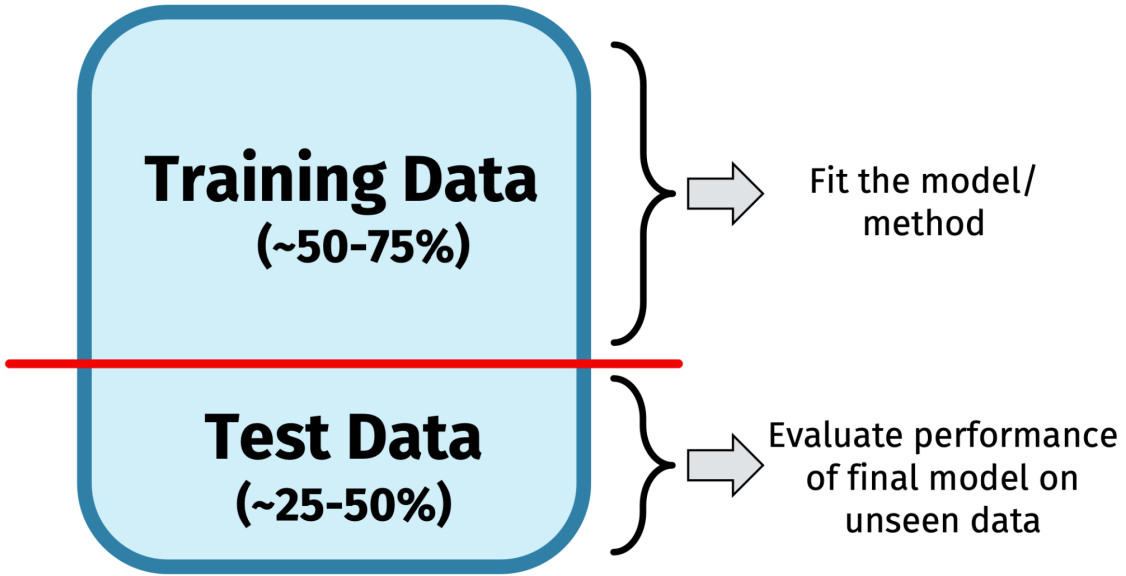
Overview of Data Splitting

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?



How to choose data splitting scheme

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

- Randomly? (default inclination)

Recall primary purpose of test set: evaluate generalizability to new data

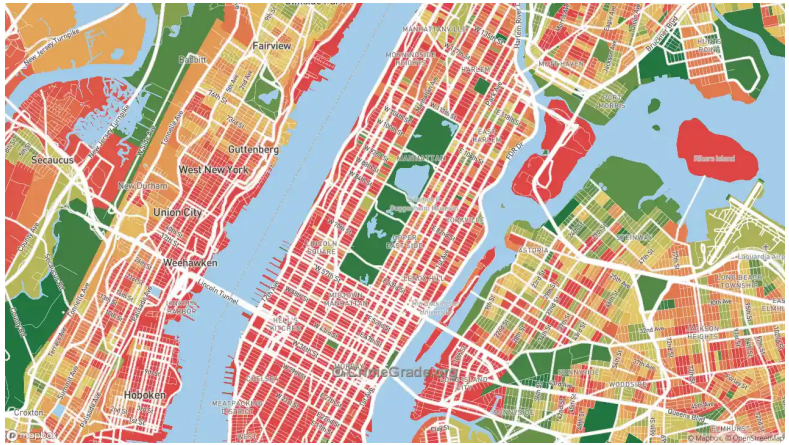
↳ **Data splitting scheme should mimic the process of obtaining this future data**

How to choose data splitting scheme

Data splitting scheme should mimic the process of obtaining new (future) data

Examples:

+ **Spatial data:** Suppose we want to predict neighborhood crime rates and have data from NYC but want to apply/generalize model to LA

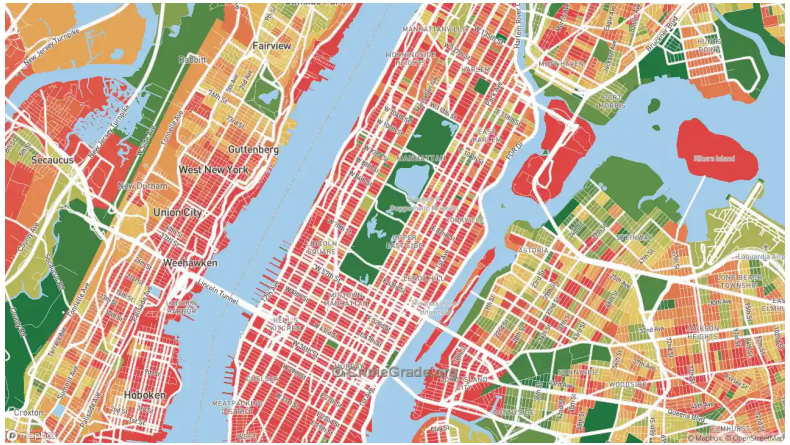


How to choose data splitting scheme

Data splitting scheme should mimic the process of obtaining new (future) data

Examples:

+ **Spatial data:** Suppose we want to predict neighborhood crime rates and have data from NYC but want to apply/generalize model to LA



- Random splitting does not mimic the process of obtaining data from a *completely new region*
+ Need to **spatially block** data and put each block fully in train or fully in test

How to choose data splitting scheme

Data splitting scheme should mimic the process of obtaining new (future) data

Examples:

+ **Spatial data:** Suppose we want to predict neighborhood crime rates and have data from NYC but want to apply/generalize model to LA

- Random splitting does not mimic the process of obtaining data from a *completely new region*
+ Need to **spatially block** data and put each block fully in train or fully in test

How to choose data splitting scheme

Data splitting scheme should mimic the process of obtaining this future data

Q: What would happen if we performed data splitting with random sampling and ignored the underlying process of obtaining future data?

Our estimated test error would be too optimistic

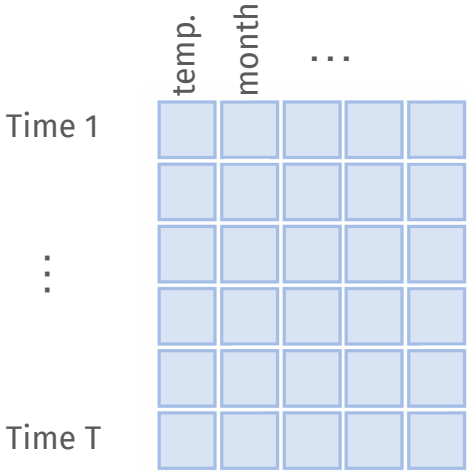
- "Our results are too good to be true"

How to choose data splitting scheme

Data splitting scheme should mimic the process of obtaining new (future) data

Examples:

+ **Temporal (time) data:** Suppose we want to forecast K-days-ahead temperatures and we have historical data to use for training purposes

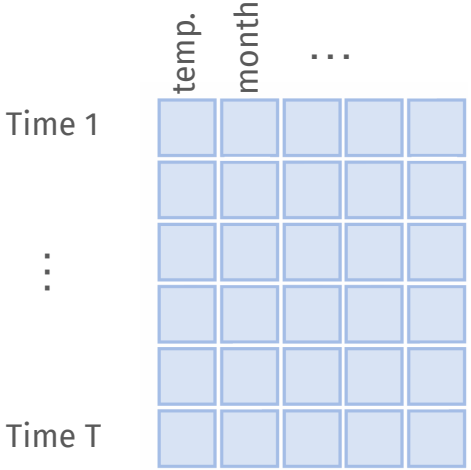


How to choose data splitting scheme

Data splitting scheme should mimic the process of obtaining new (future) data

Examples:

+ **Temporal (time) data:** Suppose we want to forecast K-days-ahead temperatures and we have historical data to use for training purposes



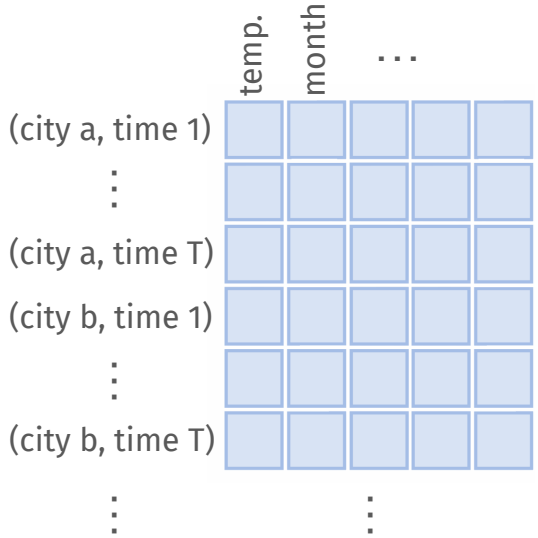
- Random splitting does not mimic the process of obtain/predicting on future temperature data
+ Use time 1 through T-K for training to predict temperature K-days ahead
    - If we need to do multiple data splits, use a "rolling window" approach

How to choose data splitting scheme

Data splitting scheme should mimic the process of obtaining new (future) data

Examples:

+ **Q:** What if we also had temperature data from different cities and wanted to generalize to existing and new cities and future time points?

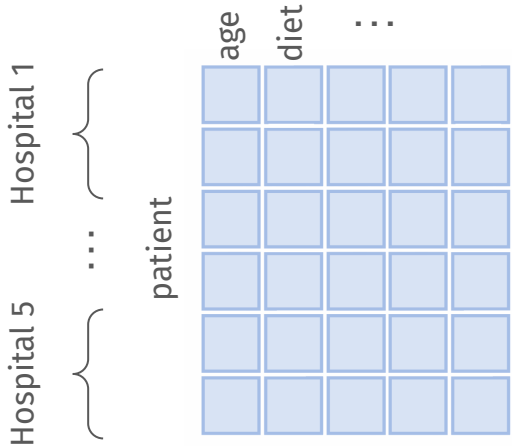


How to choose data splitting scheme

Data splitting scheme should mimic the process of obtaining new (future) data

Examples:

+ **Grouped/blocked data:** Suppose we want to develop a clinical model for risk of heart attacks and we have data from 5 hospitals

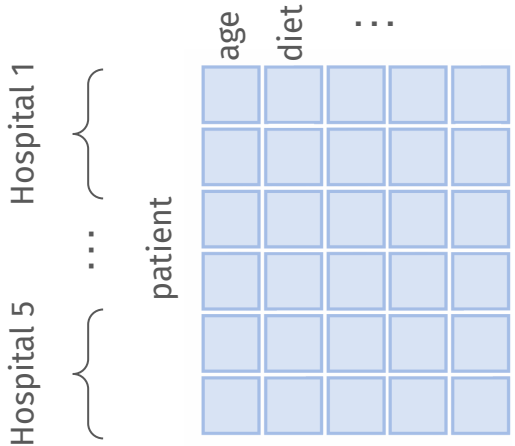


How to choose data splitting scheme

Data splitting scheme should mimic the process of obtaining new (future) data

Examples:

+ **Grouped/blocked data:** Suppose we want to develop a clinical model for risk of heart attacks and we have data from 5 hospitals



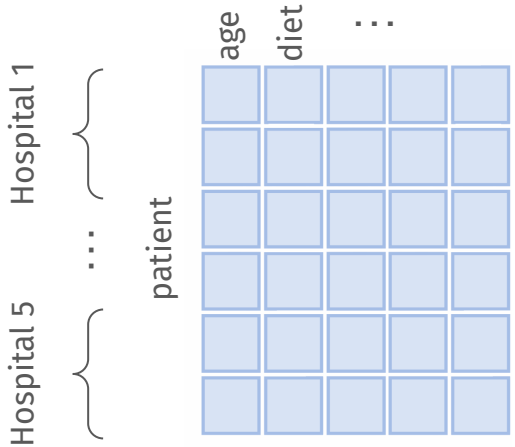
+ If we want to deploy our model just for those 5 hospitals, random splitting is ok

How to choose data splitting scheme

Data splitting scheme should mimic the process of obtaining new (future) data

Examples:

+ **Grouped/blocked data:** Suppose we want to develop a clinical model for risk of heart attacks and we have data from 5 hospitals



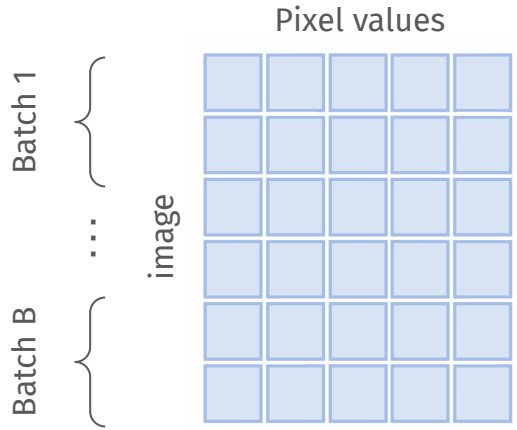
+ If we want to deploy our model just for those 5 hospitals, random splitting is ok
+ If we want to deploy our model to completely new hospitals, we should reserve at least one full hospital for the test set

How to choose data splitting scheme

Data splitting scheme should mimic the process of obtaining new (future) data

Examples:

+ **Grouped/blocked data:** Suppose we want to develop classify astronomical images from a particular telescope (or astronomer or experimental "batch")

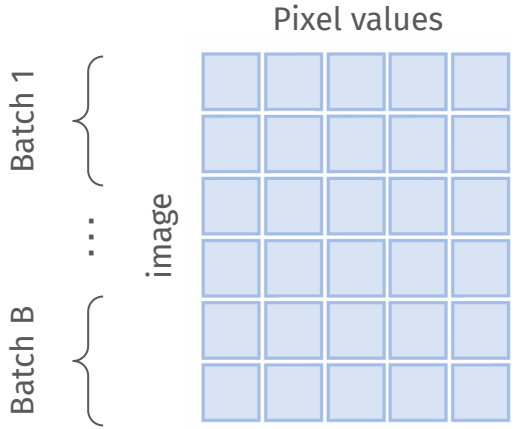


How to choose data splitting scheme

Data splitting scheme should mimic the process of obtaining new (future) data

Examples:

+ **Grouped/blocked data:** Suppose we want to develop classify astronomical images from a particular telescope (or astronomer or experimental "batch")



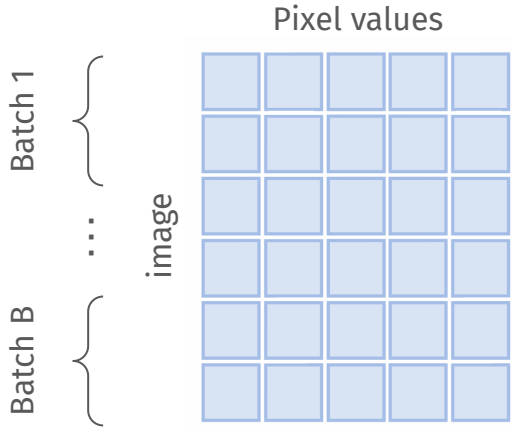
+ If we want to deploy our model just for that telescope/astronomer, random splitting is ok

How to choose data splitting scheme

Data splitting scheme should mimic the process of obtaining new (future) data

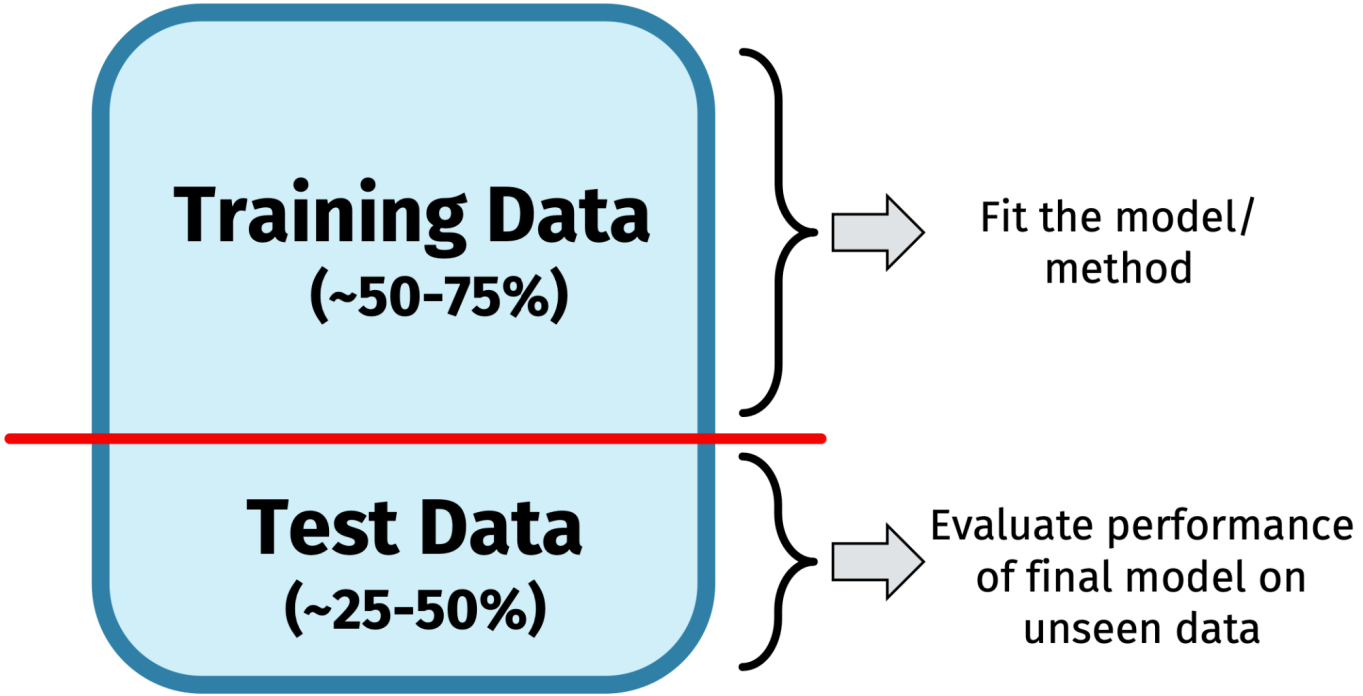
Examples:

+ **Grouped/blocked data:** Suppose we want to develop classify astronomical images from a particular telescope (or astronomer or experimental "batch")

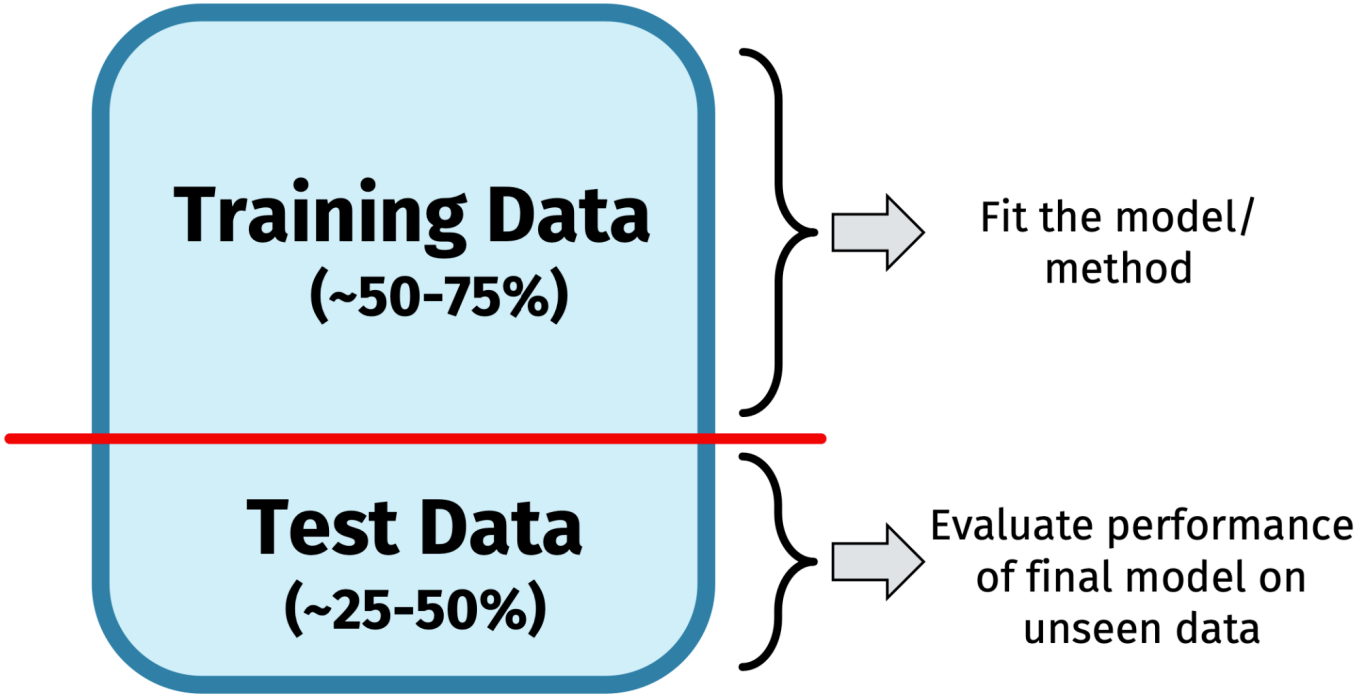


+ If we want to deploy our model just for that telescope/astronomer, random splitting is ok
+ If we want to deploy our model to a completely new telescope/astronomer, we should reserve at least one full "batch" for the test set

Data splitting: the simplest case

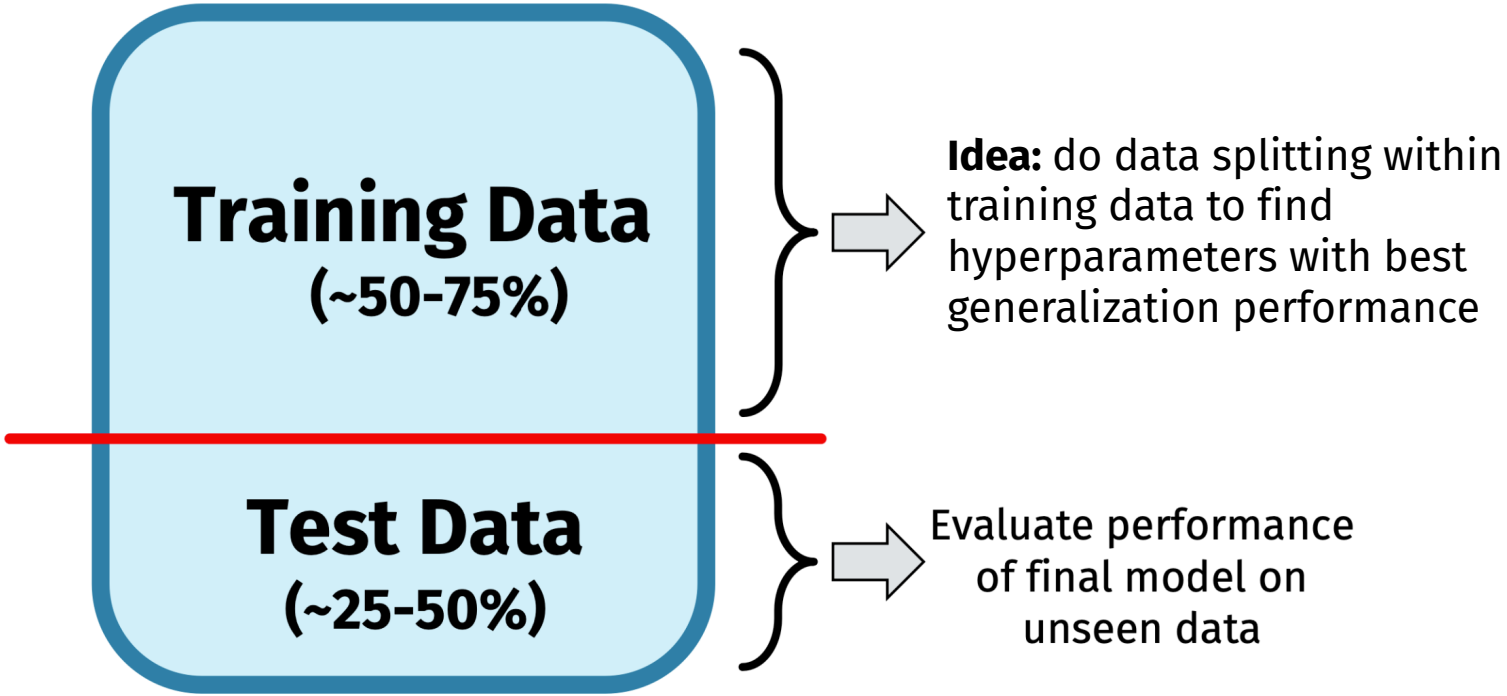


Data splitting: the simplest case

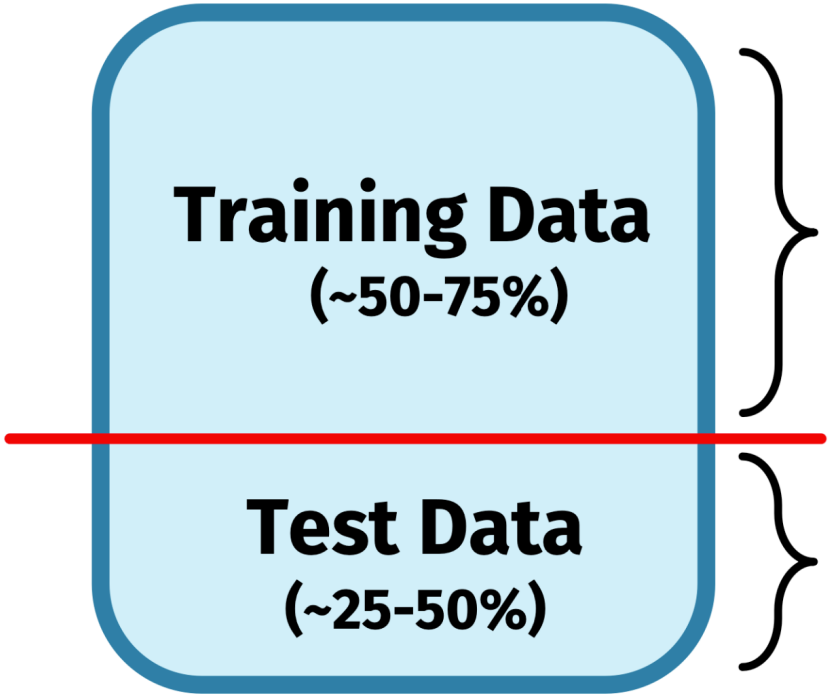


How can we modify this data splitting if we need to tune hyperparameters?

Data splitting with hyperparameter tuning



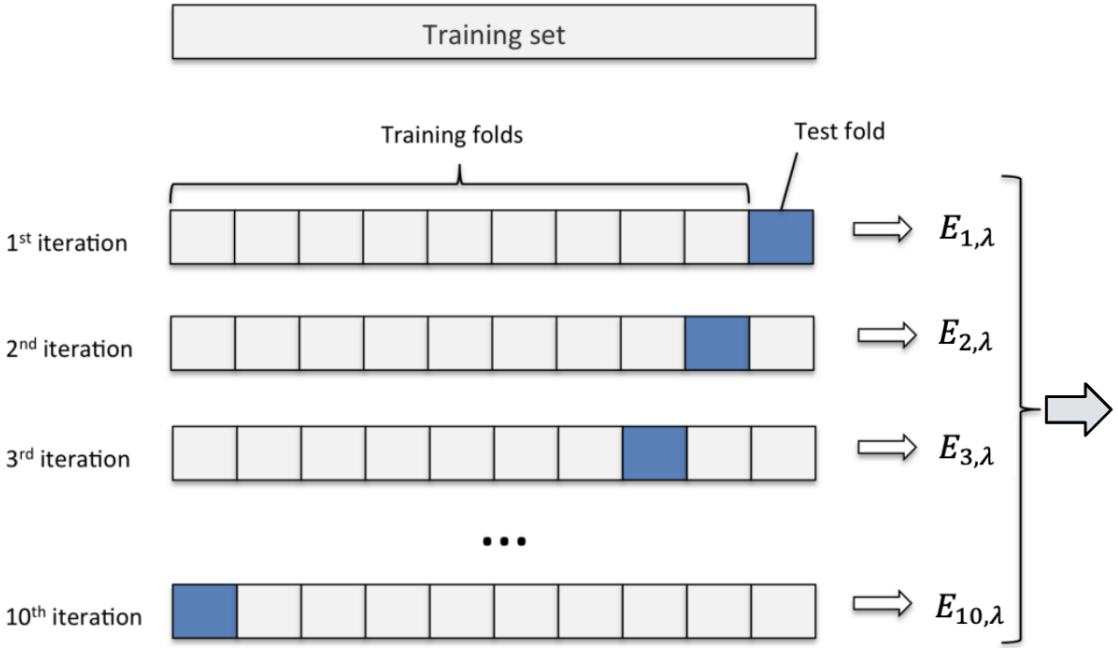
Data splitting with hyperparameter tuning

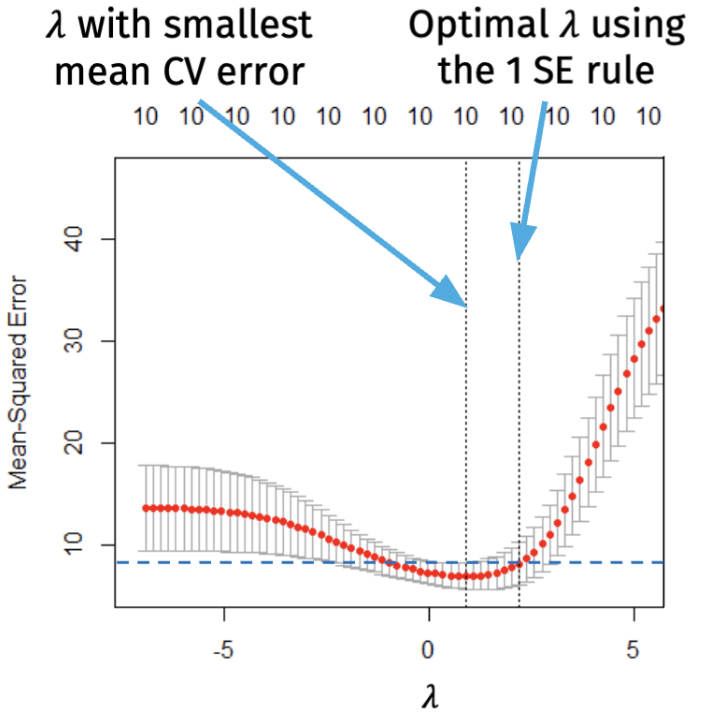


1. Do **K-fold CV** to choose hyperparameters
2. Re-fit model on full training data using optimal hyperparameters

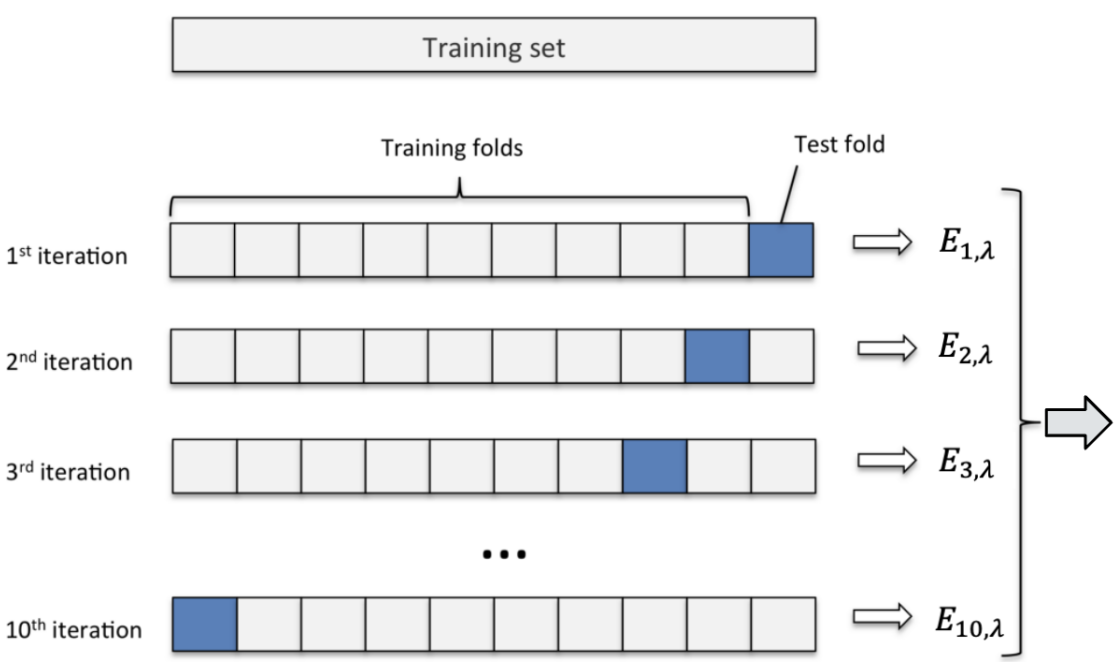
Evaluate performance of final model on unseen data

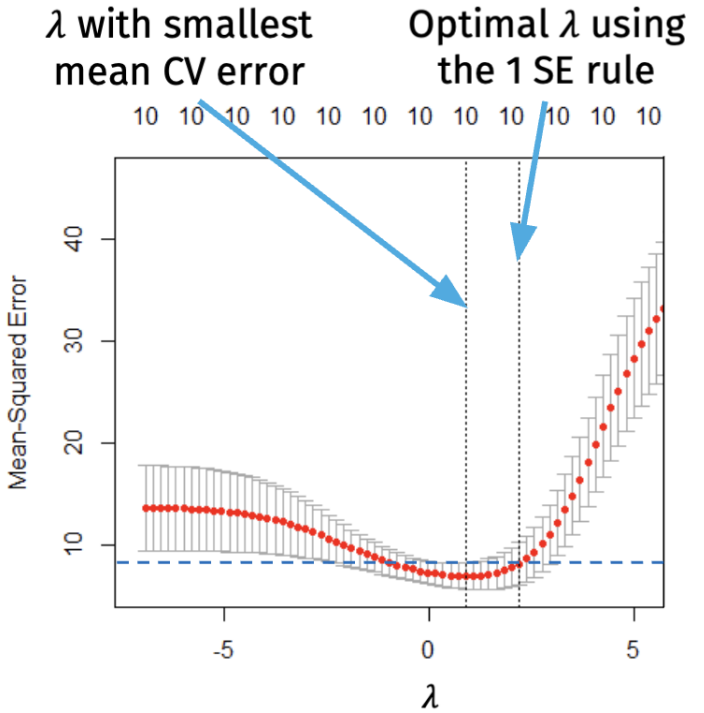
K-fold Cross-Validation (CV) for choosing hyperparameters





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

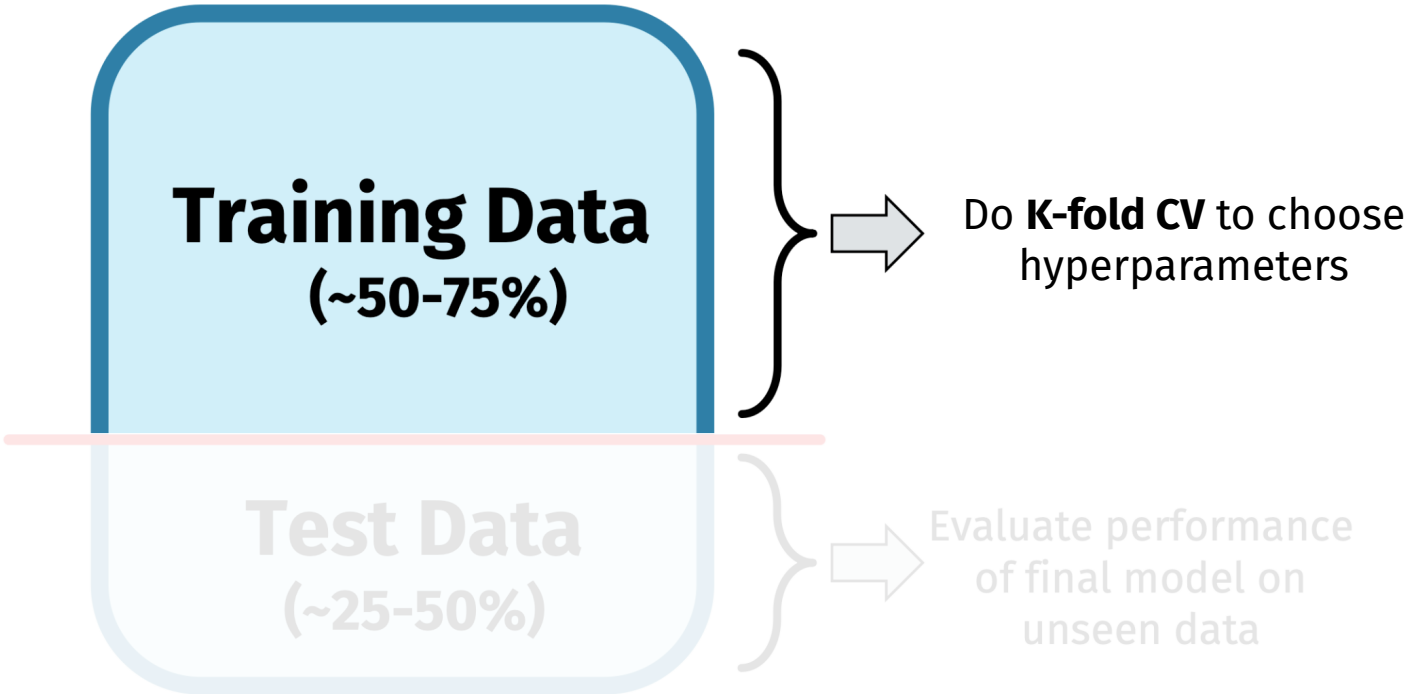




How do we split the data into folds? Typically similar to how we split into train vs test

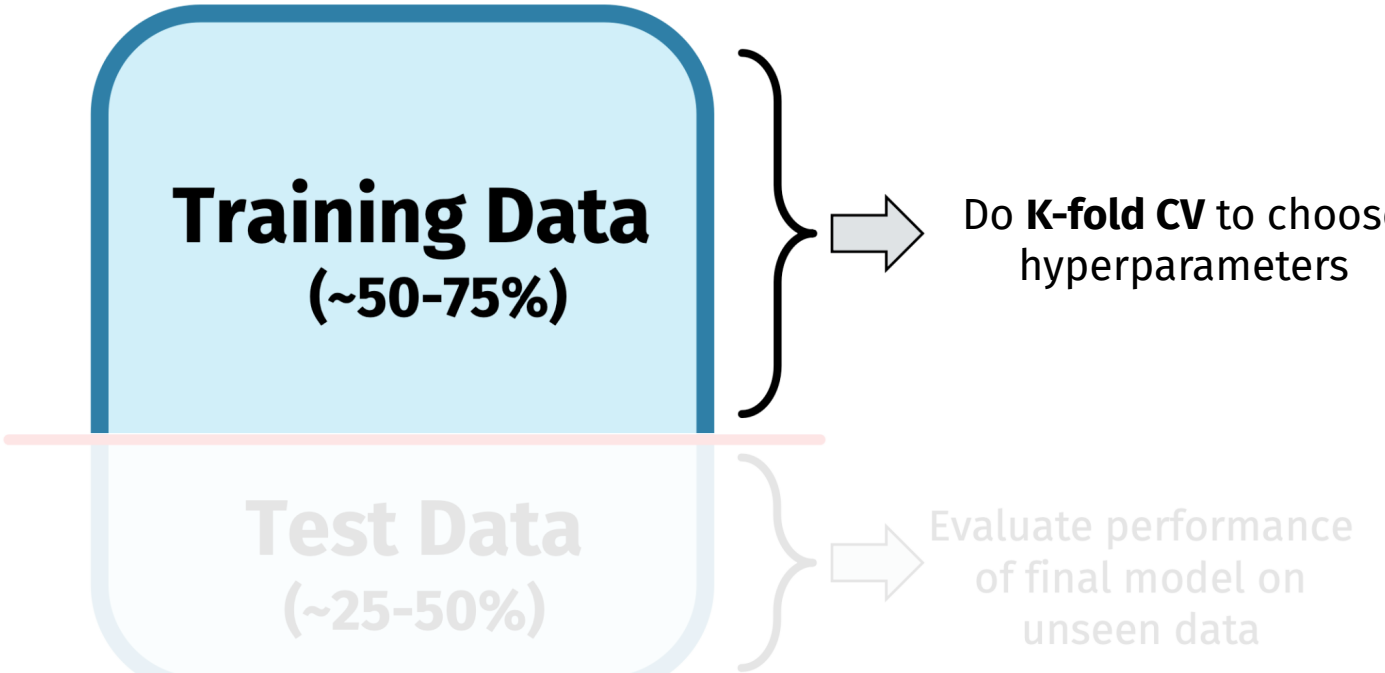
Data splitting with hyperparameter tuning

What if we report the CV error for the best hyperparameter as our test error?



Data splitting with hyperparameter tuning

What if we report the CV error for the best hyperparameter as our test error?

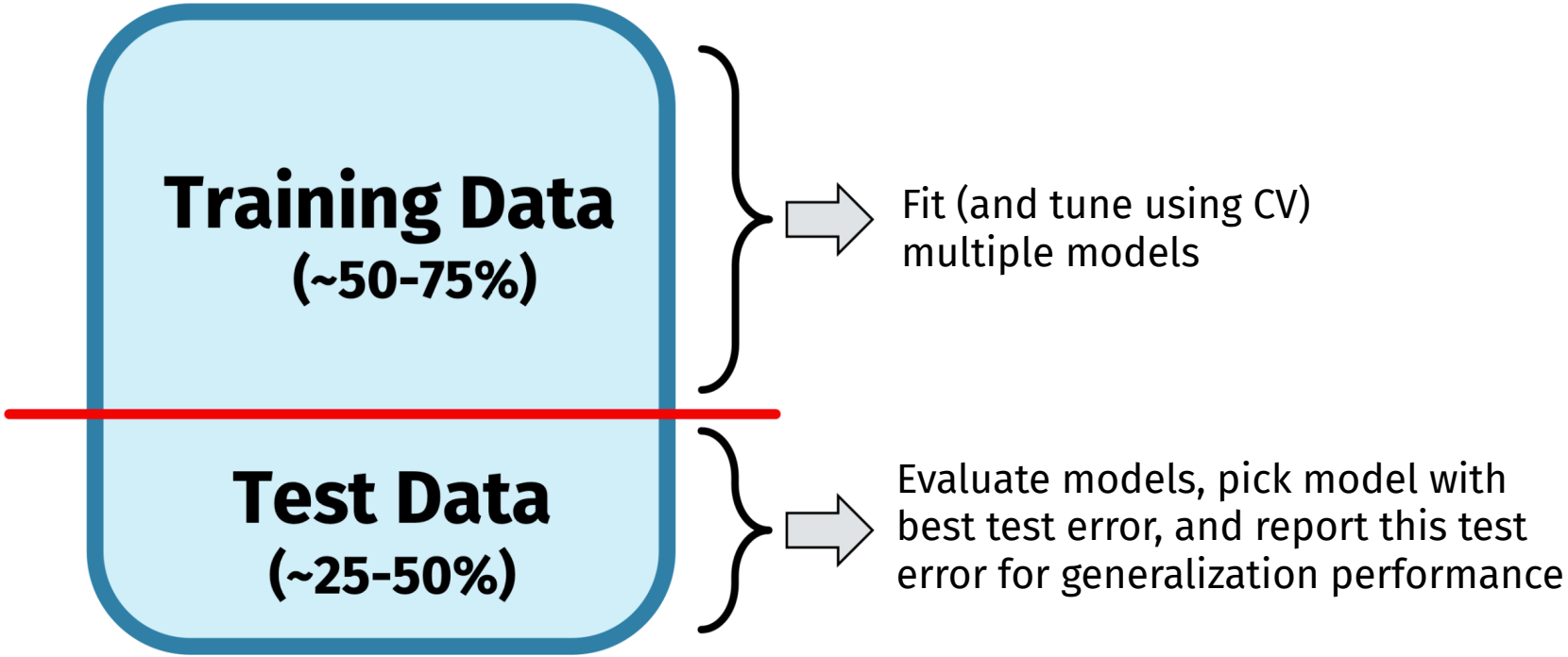


This would be an **overly-optimistic estimate** because we essentially took the minimum (or best) error across many attempts ("too good to be true")

Data splitting with hyperparameter tuning + model selection

(multiple models to choose from)

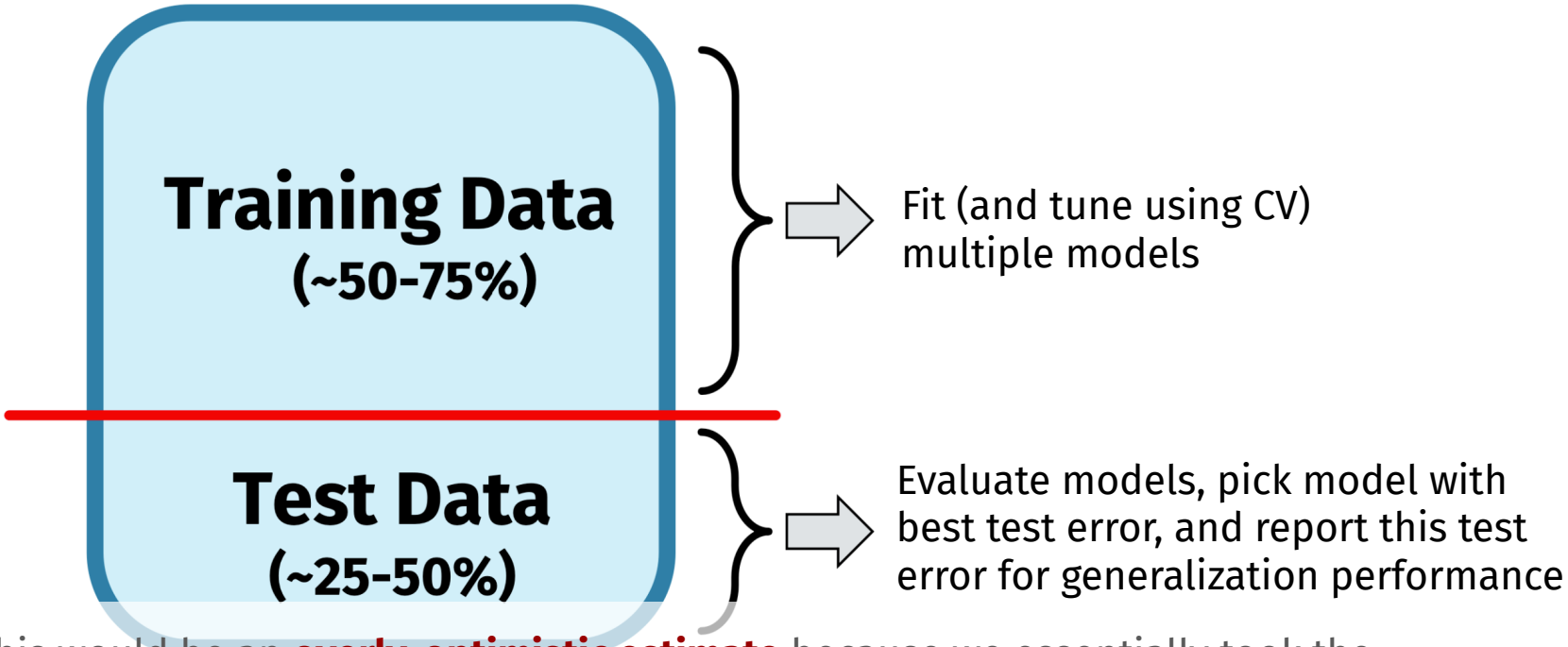
What if we report the test error after using test data to do model selection?



Data splitting with hyperparameter tuning + model selection

(multiple models to choose from)

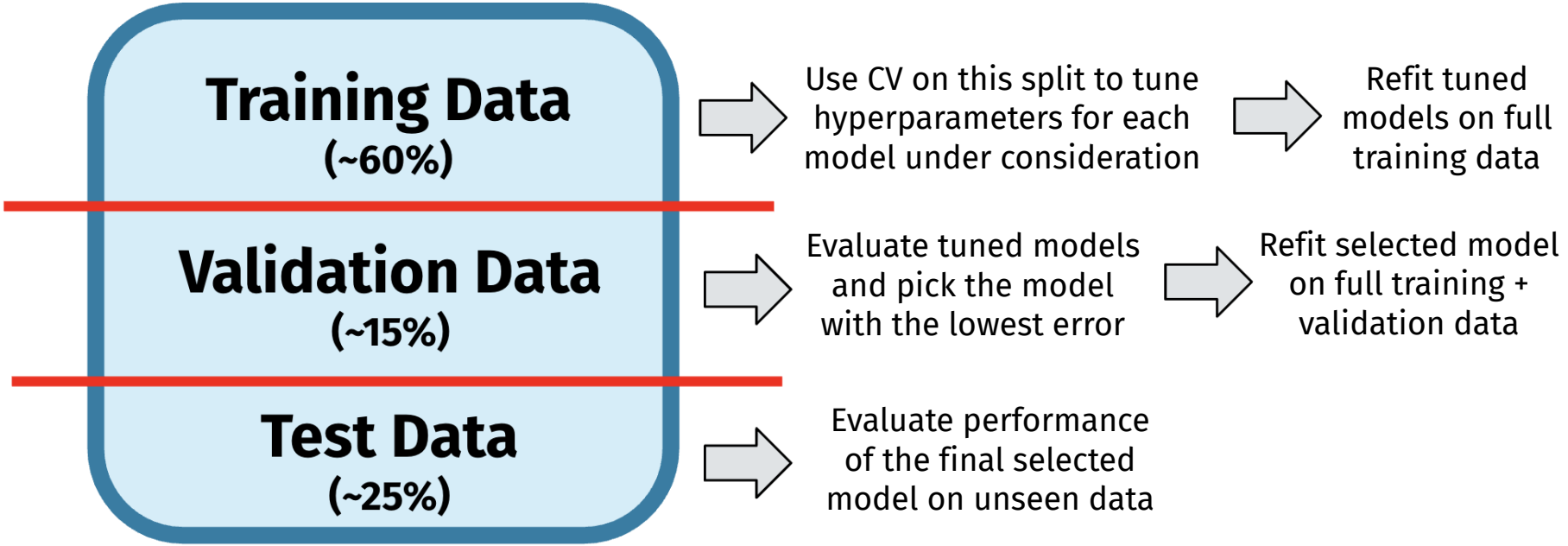
What if we report the test error after using test data to do model selection?



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")

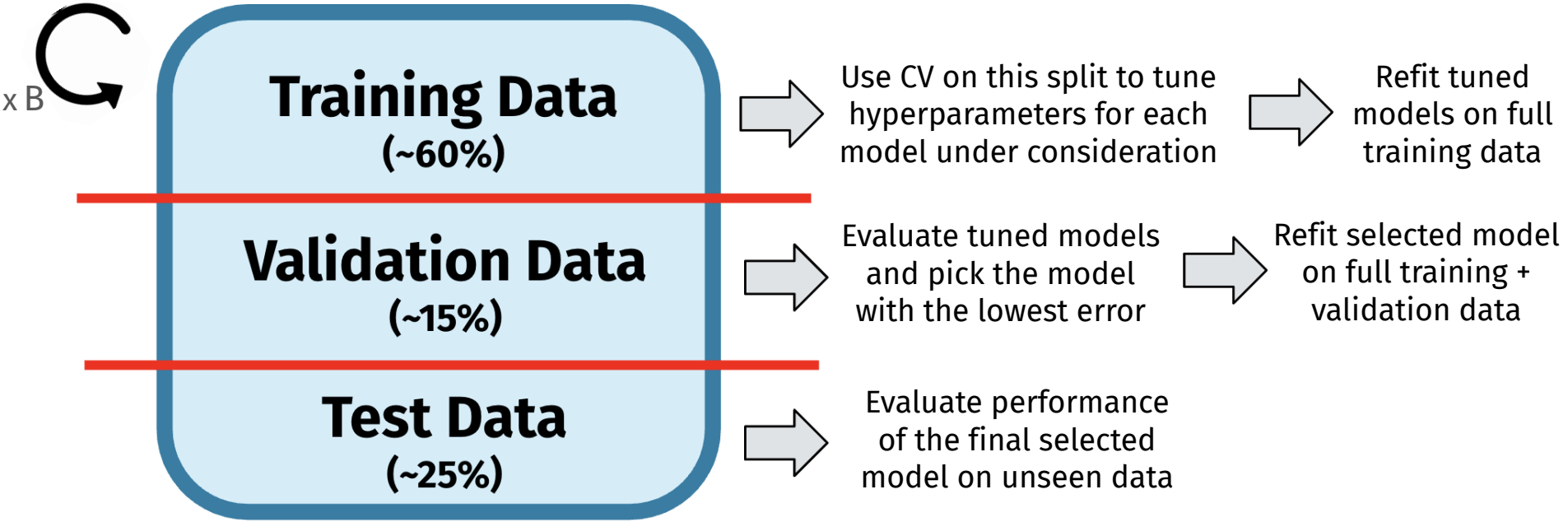
Data splitting with hyperparameter tuning + model selection

(multiple models to choose from)



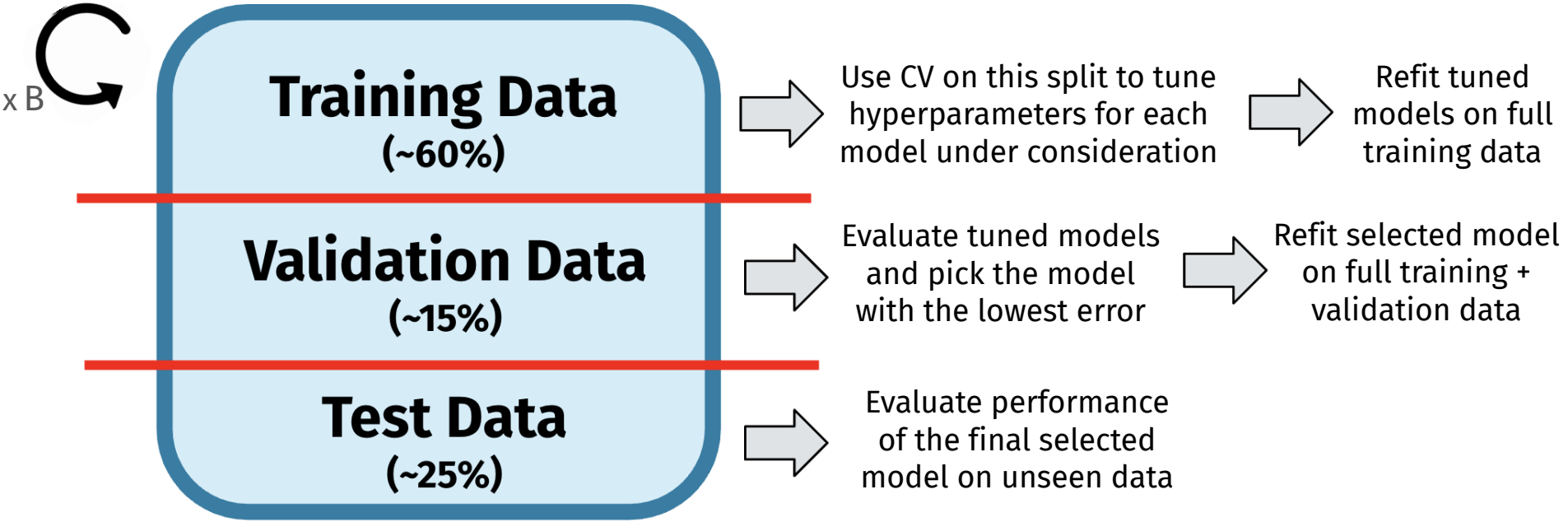
Data splitting with hyperparameter tuning + model selection

(multiple models to choose from)



Data splitting with hyperparameter tuning + model selection

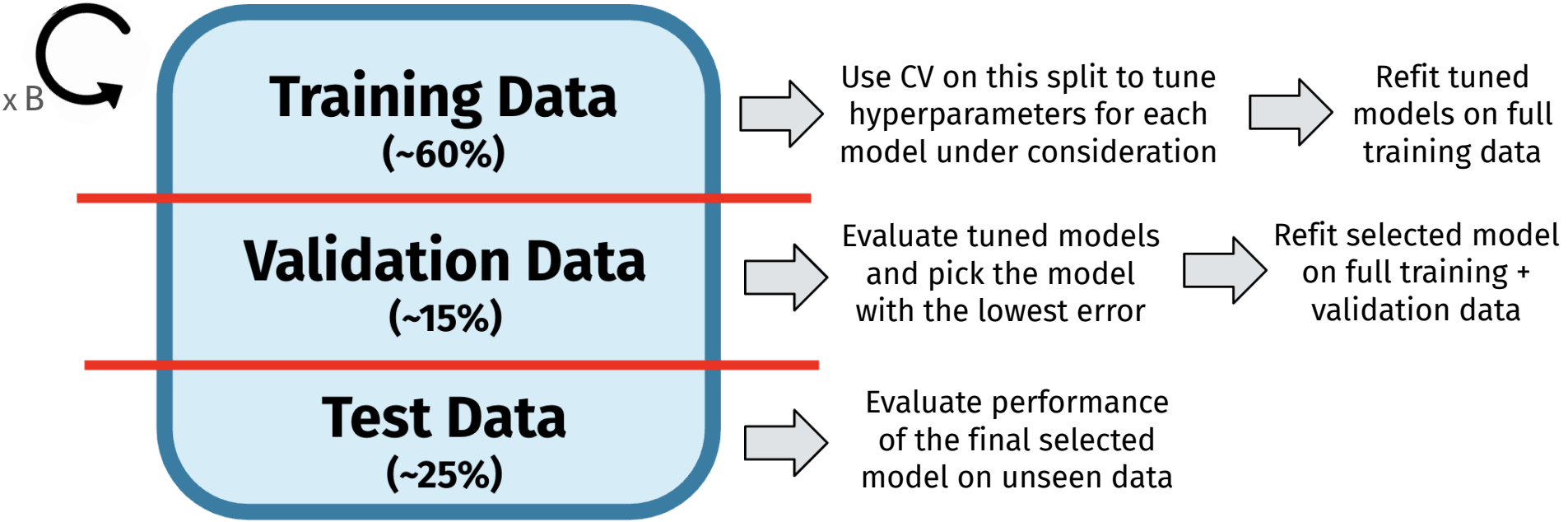
(multiple models to choose from)



+ 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**

Key Takeaways of Data Splitting

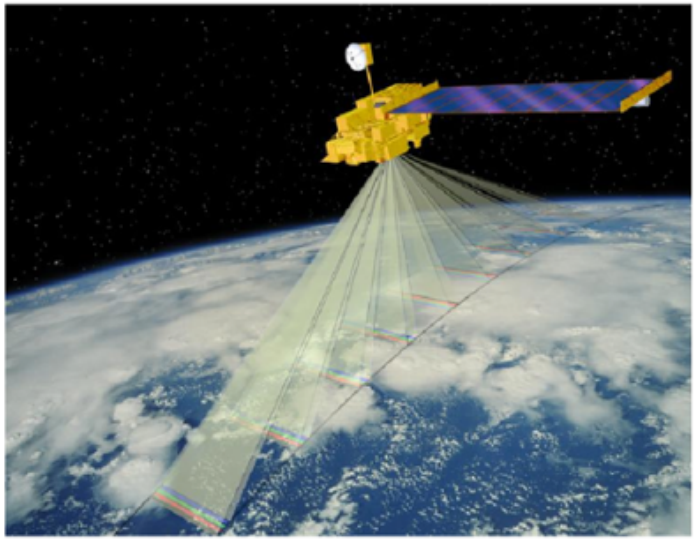
Data splitting is the key to assessing generalizability

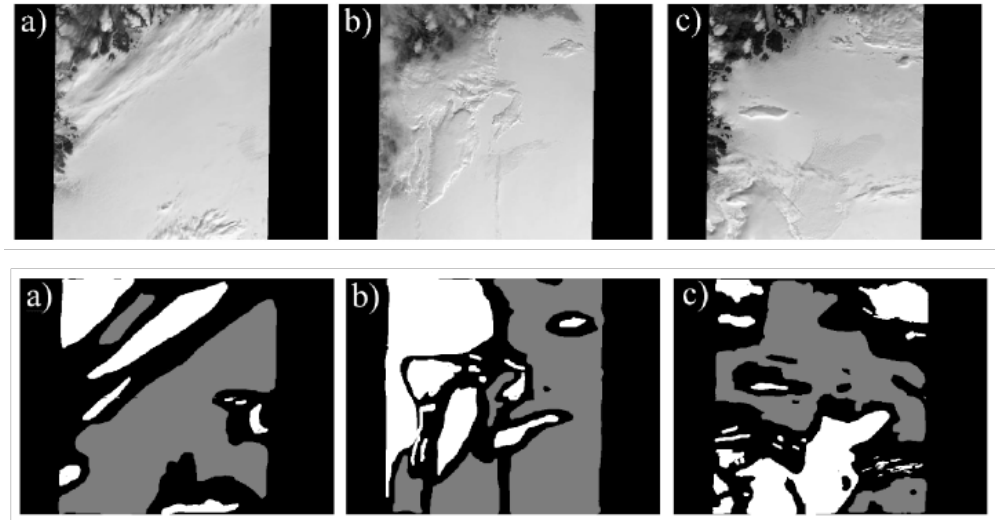
+ Need to decide how to (i) allocate samples into splits and (ii) splitting scheme
  (i) Should mimic the process of obtaining new data in the future
  (ii) Use training-validation-test split if selecting among multiple models;
  Use training-test if considering only one type of model
+ **Purpose of 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

Introduction to Lab 3

Lab 3: Remote sensing for cloud detection (due March 6 6pm)

Goal: predict whether each pixel is a cloud or ice (from a glacier)





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