Machine Learning in R

Lecture 10 — Updating risk prediction models

Jean Feng— Epidemiology and Biostatistics

Today's lecture

1. Motivation
2. Distributional shifts
3. Model recalibration
4. Model revision
5. Selecting the best update
6. Model-specific update methods
7. Online model updating

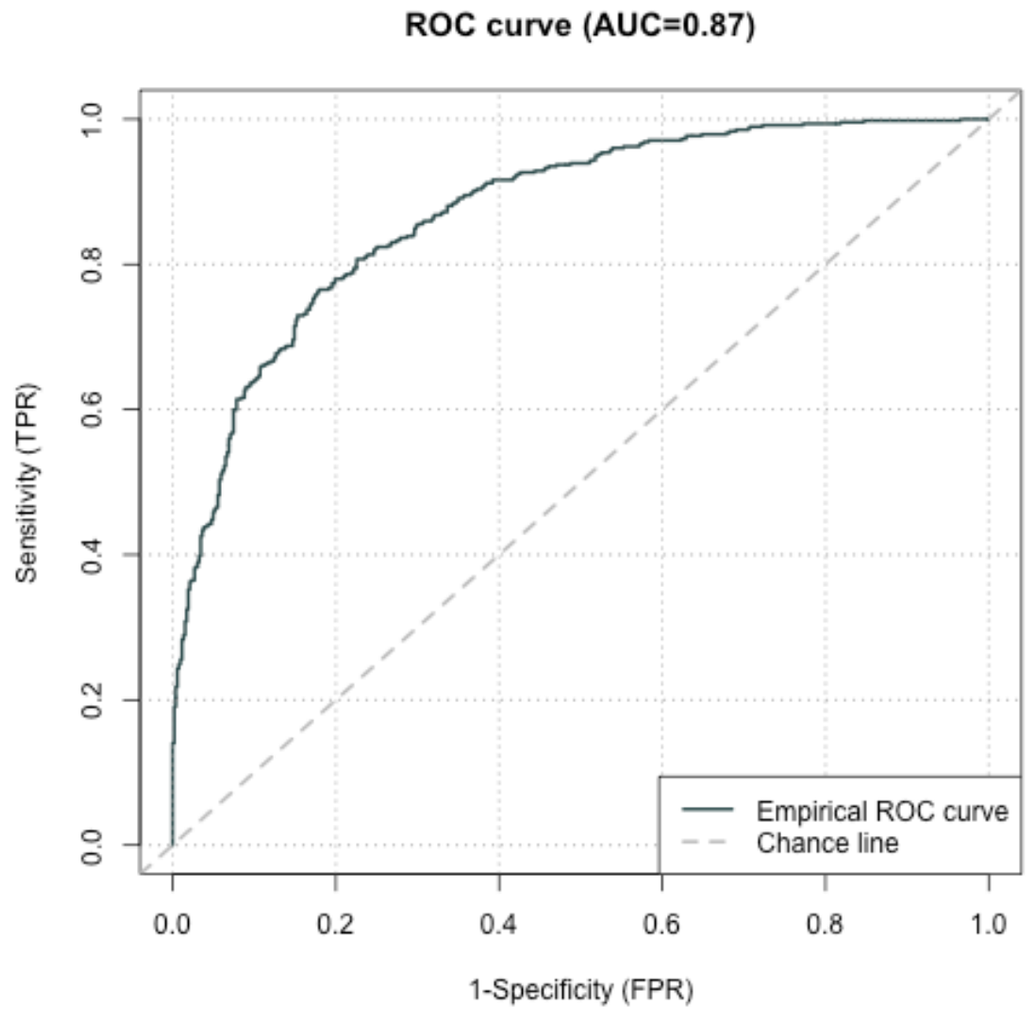
Motivation: COPD prediction

- Suppose you've developed a risk prediction model to identify patients at high risk of developing chronic obstructive pulmonary disease (COPD) based on the population at UCSF.

- You've fit a logistic regression model with the predictors:
  - Age
  - Sex
  - Smoking history in pack-years
  - Peak expiratory flow rate
  - …

Motivation: COPD prediction

- You validate your model on a prospective dataset. Success!



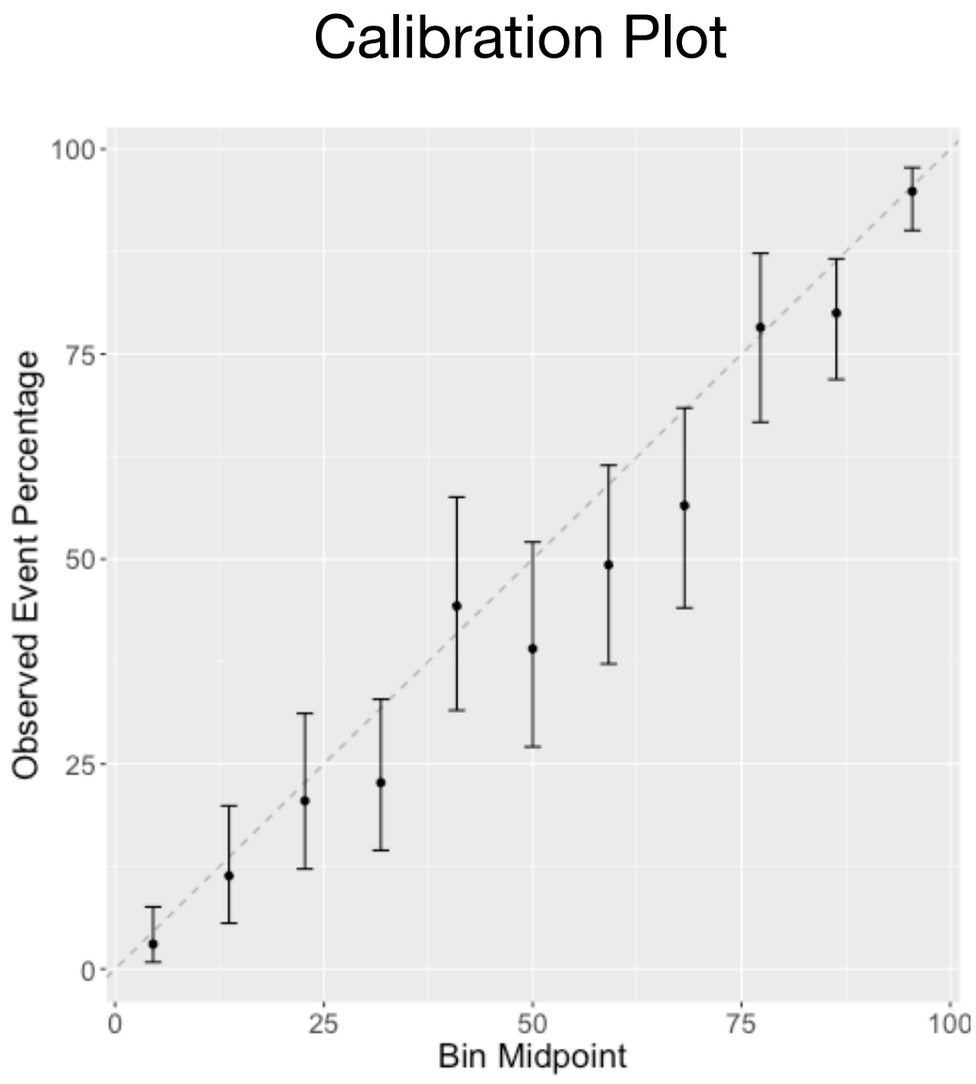
Motivation: COPD prediction

- You validate your model on a prospective dataset. Success!

Calibration measures agreement between observed outcomes and predictions.

A model is calibrated if we observe a p% event rate among patients for whom the model predicts a risk of p%.

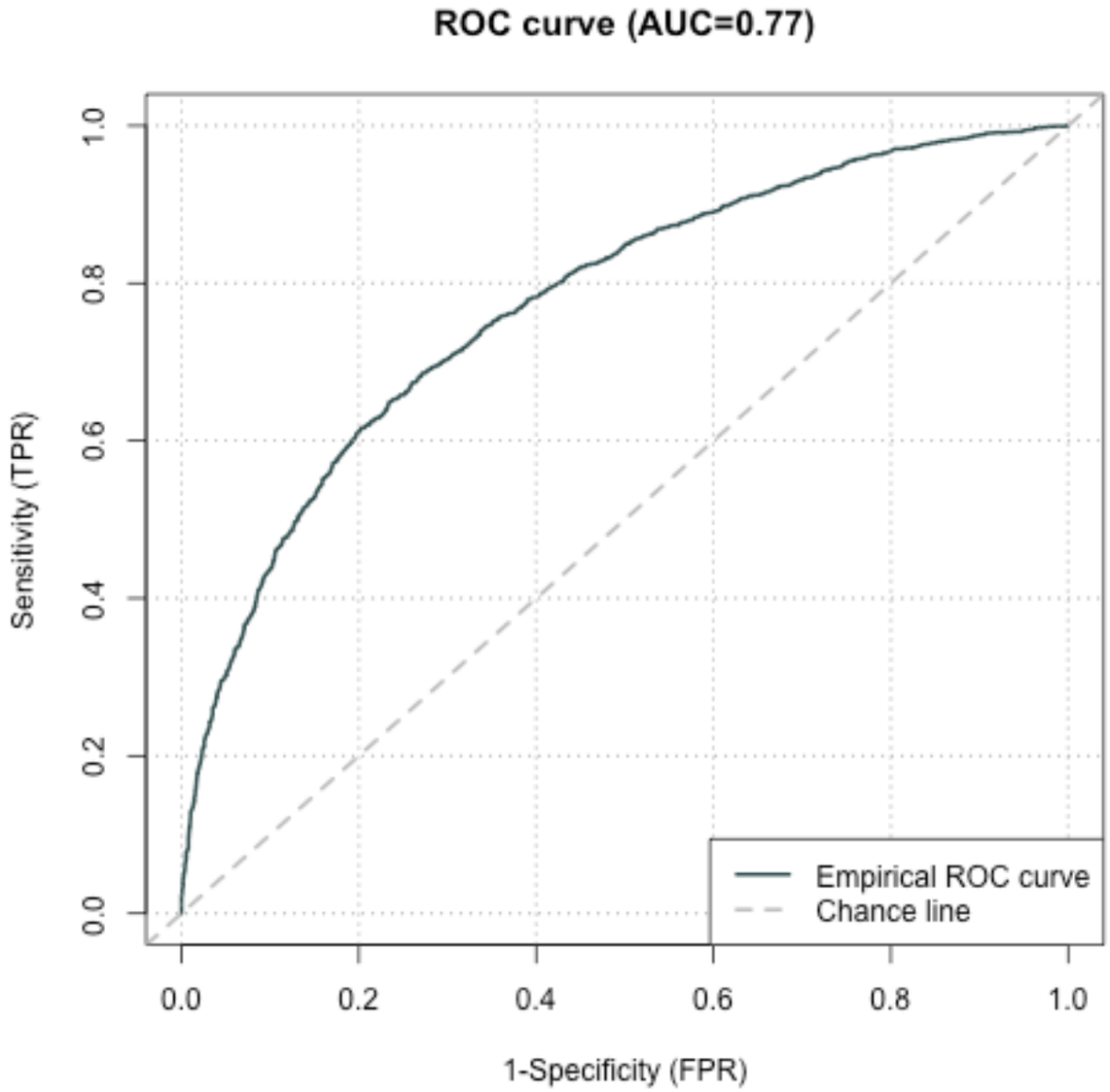
Also referred to as “reliability.”

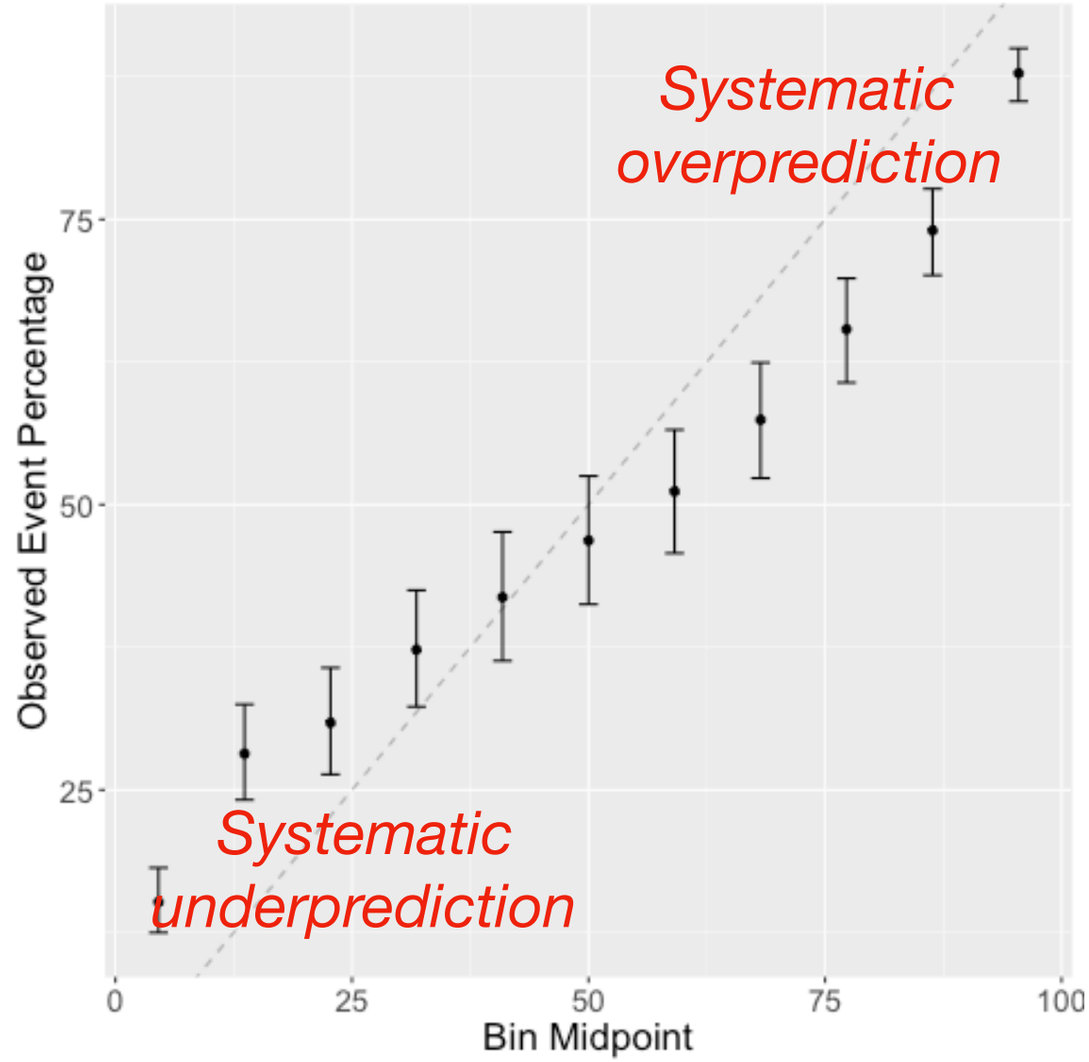


Motivation: COPD prediction

- Given the success of your model, medical institution ABC is eager to use your model. They assemble a test dataset to validate your method.

- The results are similar, but not as good:





Why did the performance drop?

- Decays in model calibration and discrimination can occur due to ***distribution shifts*** *(or* ***dataset shifts****)*. That is, the population you are deploying the model on is different from the original training/test data.

- Sources of distributional shift:
  - Changes in patient case mix
  - Shifts in clinical practice patterns
  - Introduction of new interventions
  - …

Today's lecture

1. Motivation
2. Distributional shifts
3. Model recalibration
4. Model revision
5. Selecting the best update
6. Model-specific update methods
7. Online model updating

Types of distribution shifts

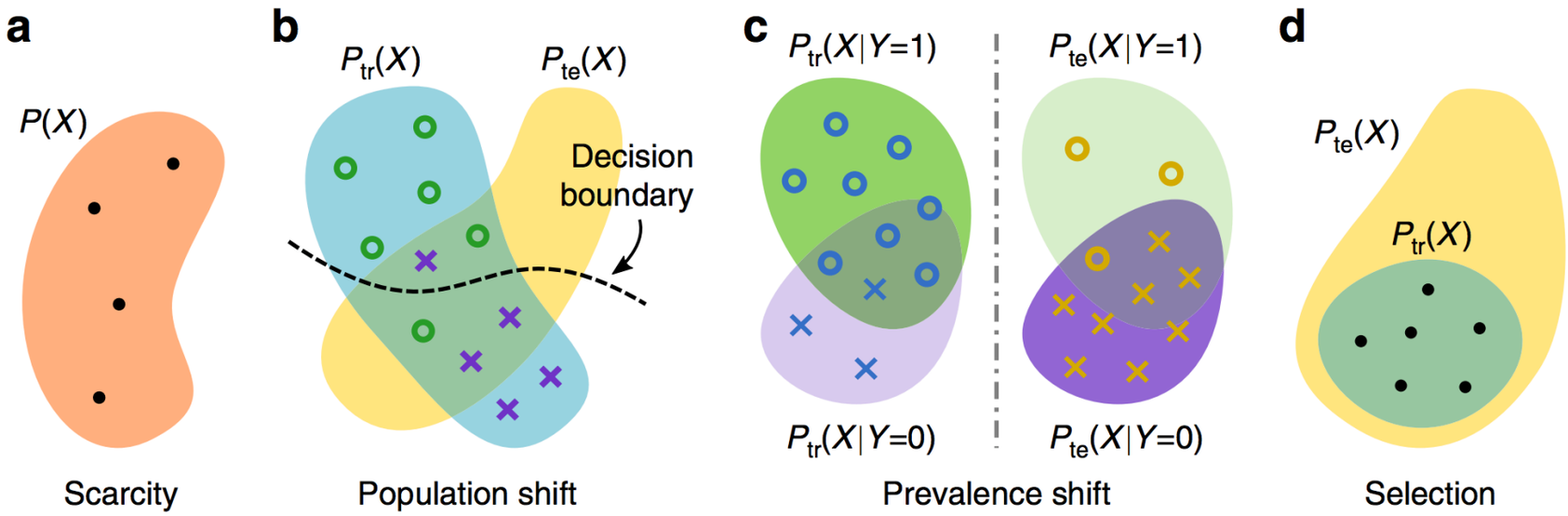


Fig. 1 Key challenges in machine learning for medical imaging. a Data scarcity and **b–d** data mismatch. X represents images and Y, annotations (e.g. diagnosis labels). P_{tr} refers to the distribution of data available for training a predictive model, and P_{te} is the test distribution, i.e. data that will be encountered once the model is deployed. Dots represent data points with any label, while circles and crosses indicate images with different labels (e.g. cases vs. controls).

Castro 2020

Covariate shift

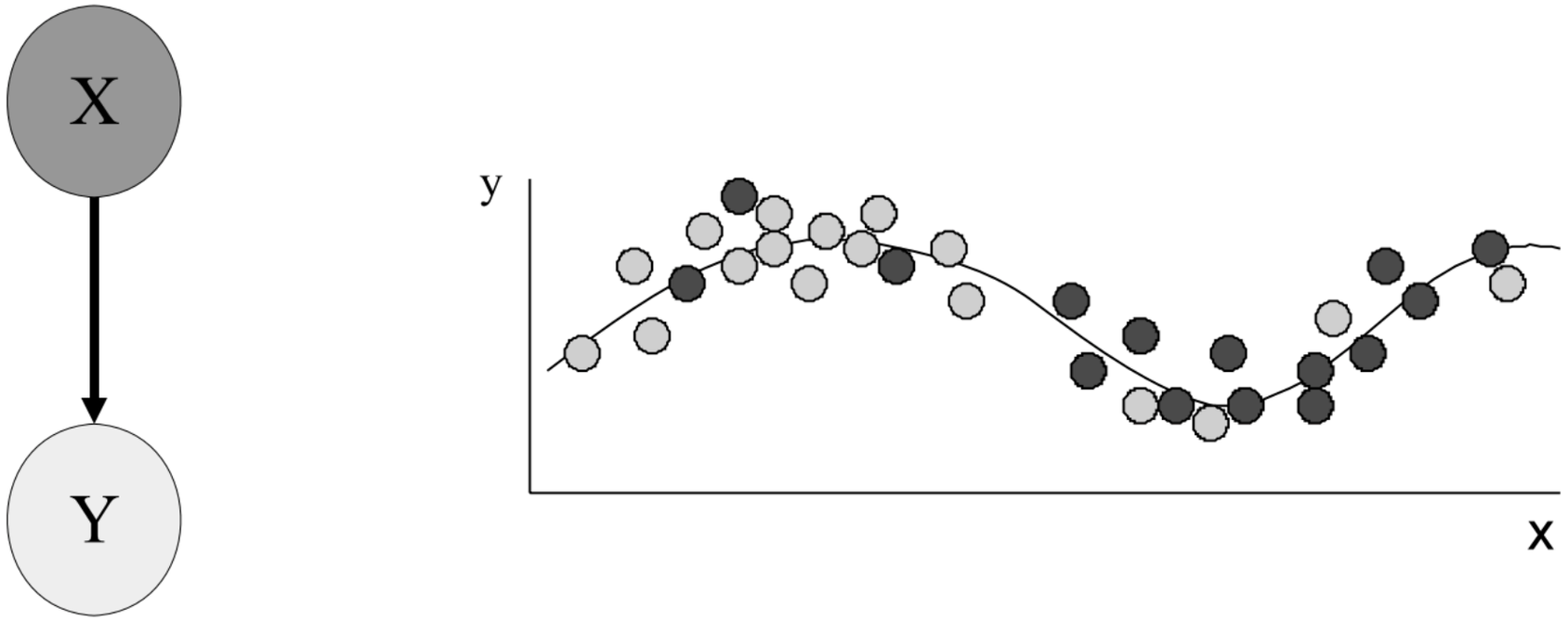


Figure 1.1 Simple covariate shift. Here the causal model indicated the targets **y** are directly dependent on the covariates **x**. In other words the predictive function and noise model stay the same, it is just the typical locations **x** of the points at which the function needs to be evaluated that change. In this figure and throughout, the causal model is given on the left with the node that varies between training and test made darker. To the right is some example data, with the training data in shaded light and the test data shaded dark.

Quinonero-Candela 2008

Prior probability shift

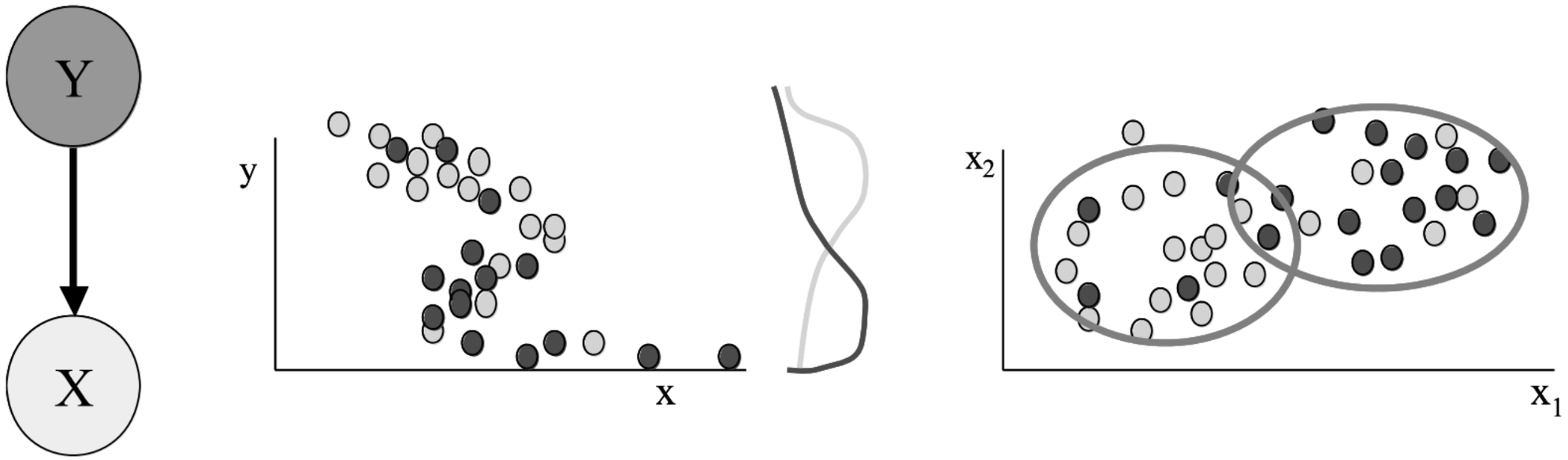


Figure 1.3 Prior probability shift. Here the causal model indicated the covariates $\mathbf{x}$ are directly dependent on the predictors $\mathbf{y}$. The distribution over $\mathbf{y}$ can change, and this effects the predictions in both the continuous case *(left)* and the class conditional case *(right)*.

Sample selection bias

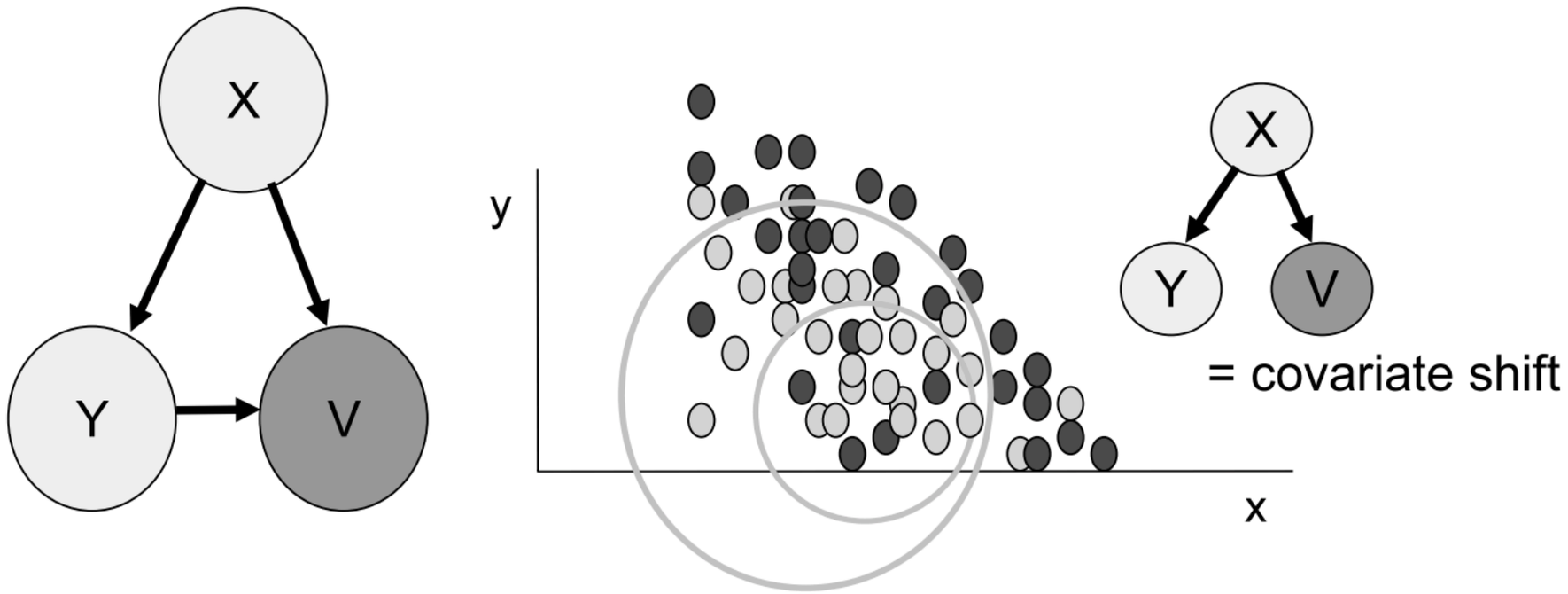


Figure 1.4 Sample selection bias. The actual observed training data is different from the test data because some of the data is more likely to be excluded from the sample. Here v denotes the selection variable, and an example selection function is given by the equiprobable contours. The dependence on $\mathbf{y}$ is crucial as without it there is no bias and this becomes a case of simple covariate shift.

Examples of real-world distribution shifts

Recalibration of the delirium prediction model for ICU patients (PRE-DELIRIC): a multinational observational study

Boogard, et. al. 2013

The PRE-DELIRIC model is a logistic regression model with 10 predictors developed and validated on data from the Netherlands that achieved an AUC of 0.85.

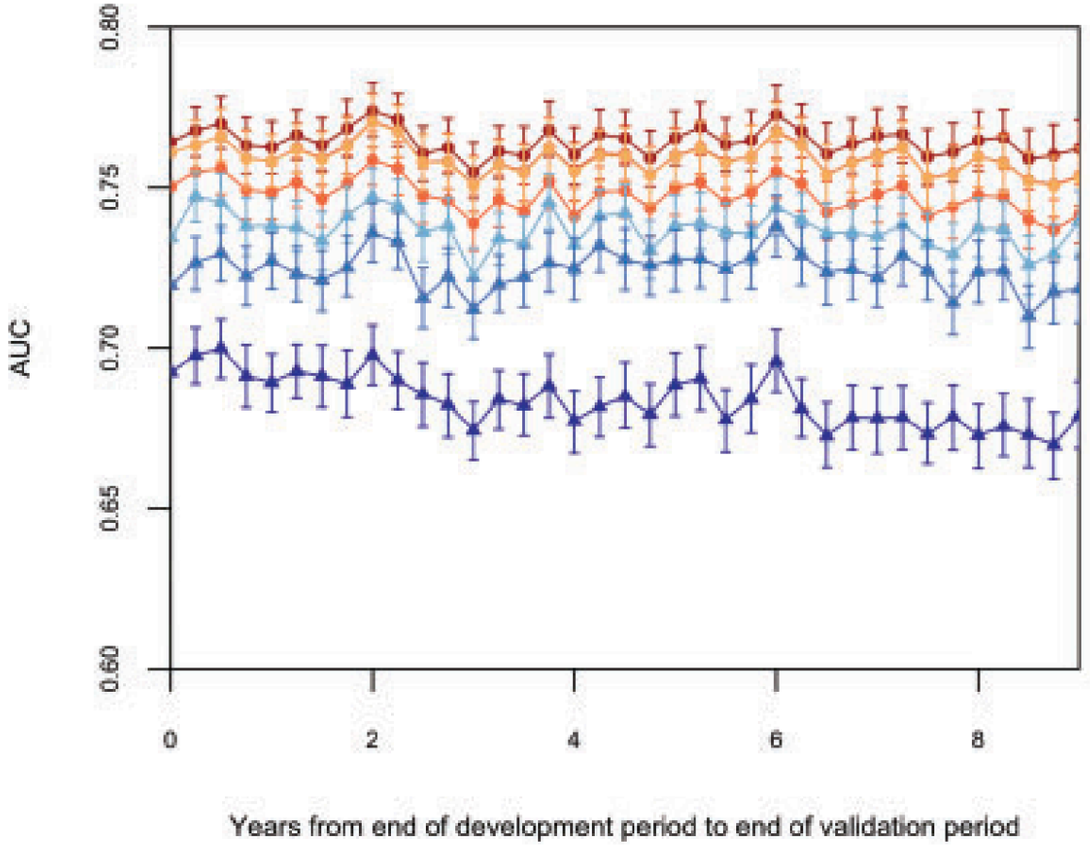
The model was evaluated in eight institutions to determine its multinational performance.

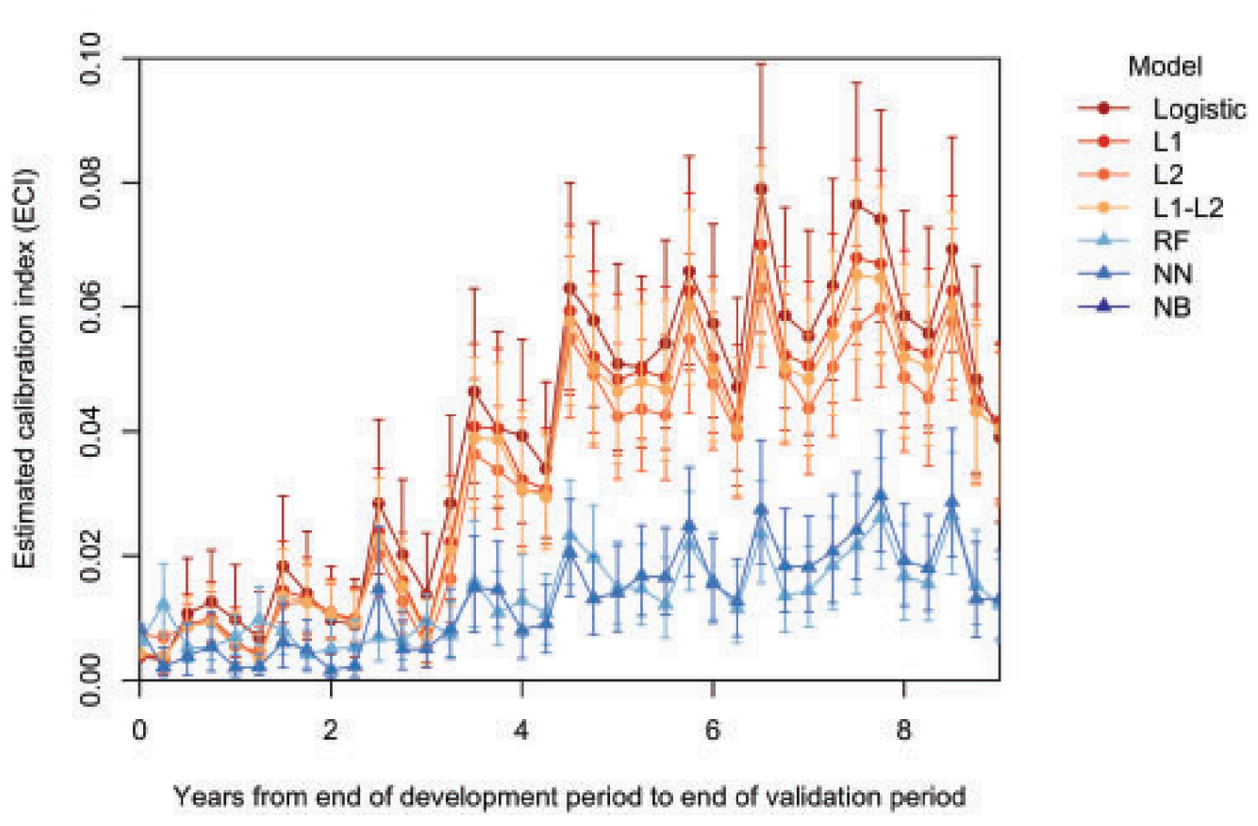
	AUROC	*95% CI*
Belgium (%)	*0.79*	0.74-0.84
Germany (%)	*0.85*	0.80-0.91
Spain (%)	*0.88*	0.81-0.94
Sweden (%)	*0.71*	0.60-0.83
Australia Brisbane (%)	*0.81*	0.75-0.88
Australia Canberra (%)	*0.80*	0.72-0.89
Australia overall	*0.81*	0.76-0.86
UK_Prescot (%)	*0.65*	0.57-0.73
UK_Kent (%)	*0.76*	0.65-0.88
UK_overall	*0.68*	0.62-0.74
PRE-DELIRIC overall	***0.77***	**0.74-0.79**

Examples of real-world distributional shifts

Calibration drift in regression and machine learning models for acute kidney injury

Sharon E Davis,[1] Thomas A Lasko,[1] Guanhua Chen,[2] Edward D Siew,[3,4] Michael E Matheny[1,2,3,5]

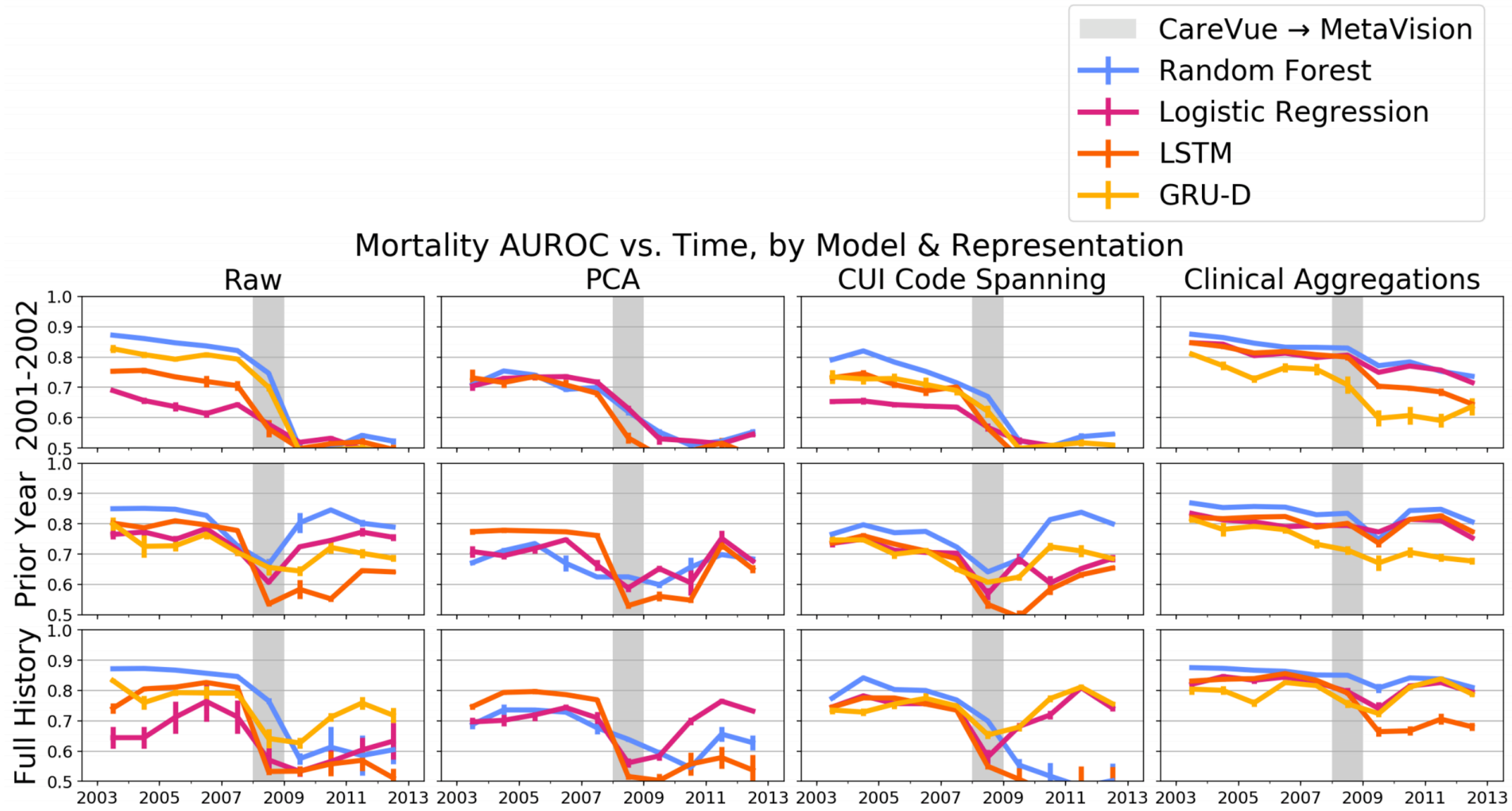




Examples of real-world distributional shifts

Feature Robustness in Non-stationary Health Records:
Caveats to Deployable Model Performance in Common Clinical Machine Learning Tasks

Nestor, et. al. 2019

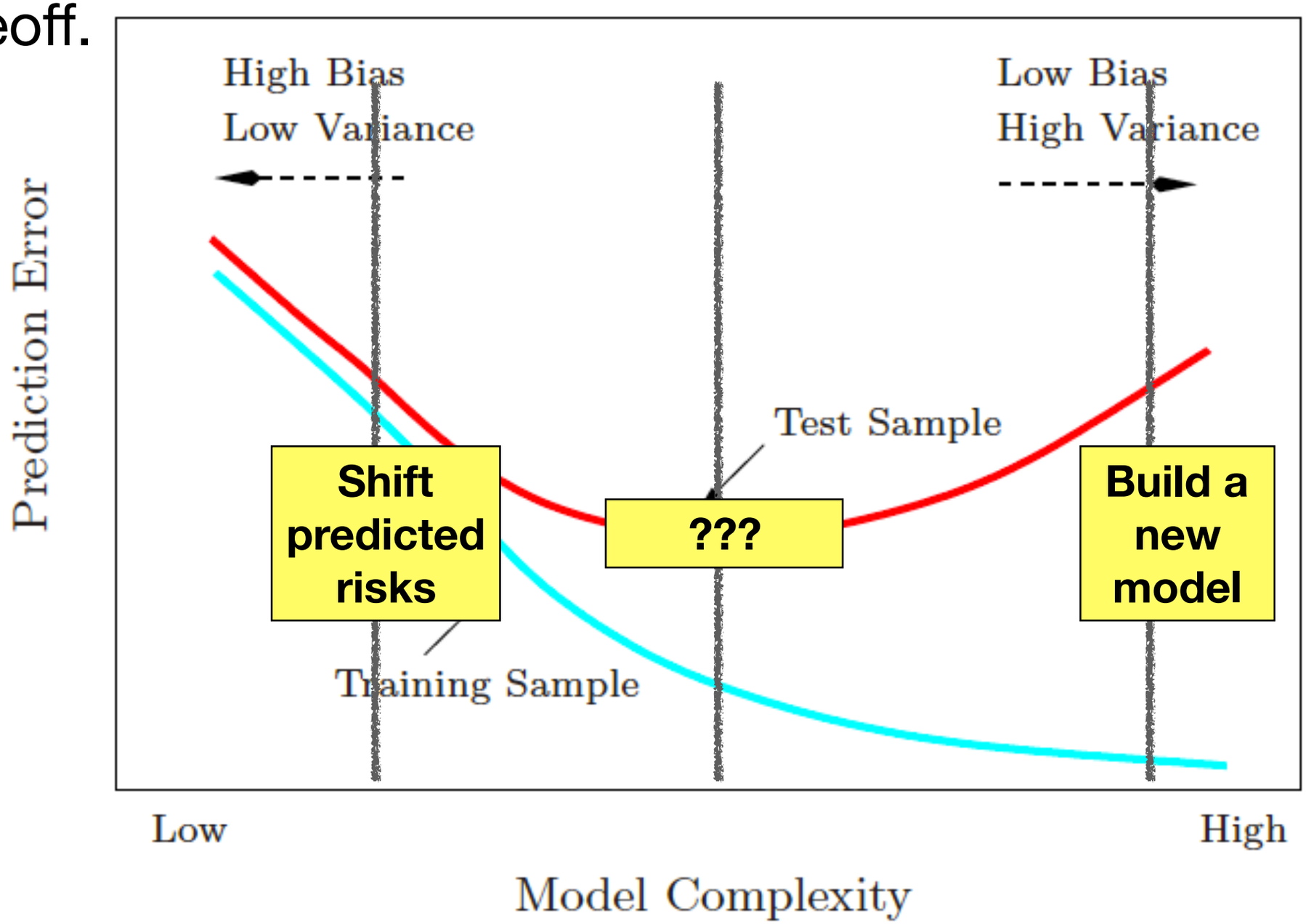


Going back to our COPD problem…

- You would like to update the risk prediction model for medical institution ABC. Methods vary in how extensive the update is:
    - Recalibration:
        - Shift the predicted risks by some constant.
        - Shift and scale the predicted risk scores.
    - Model revision
        - Do tiny updates to a subset of the model coefficients.
        - Do tiny updates to all the model coefficients.
    - Model extension: Use the original model and add new variables
    - Build a completely new model.

Bias-variance tradeoff

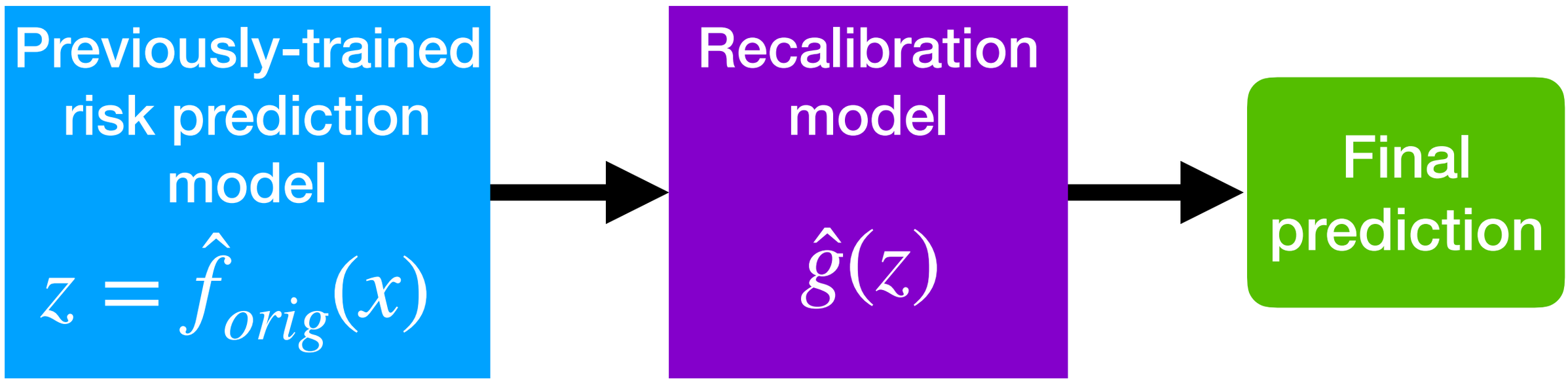
- There is typically less data from your new target population. You need to think about the bias-variance tradeoff.



Today's lecture

1. Motivation
2. Distributional shifts
3. Model recalibration
4. Model revision
5. Selecting the best update
6. Model-specific update methods
7. Online model updating

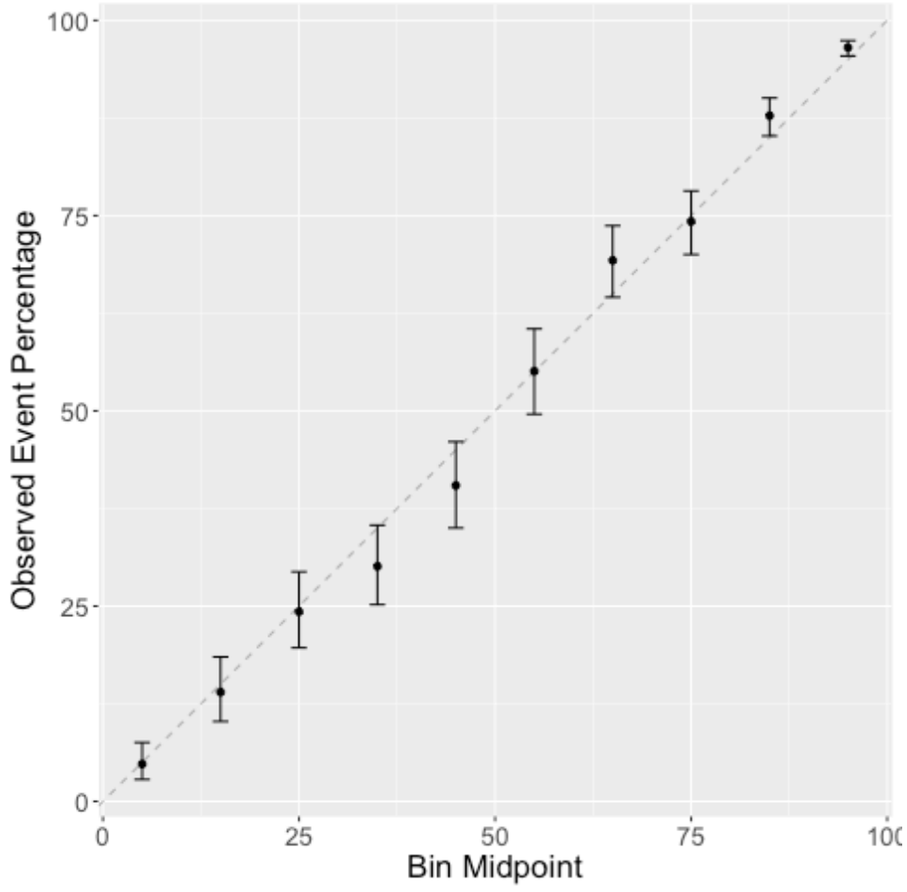
Model recalibration

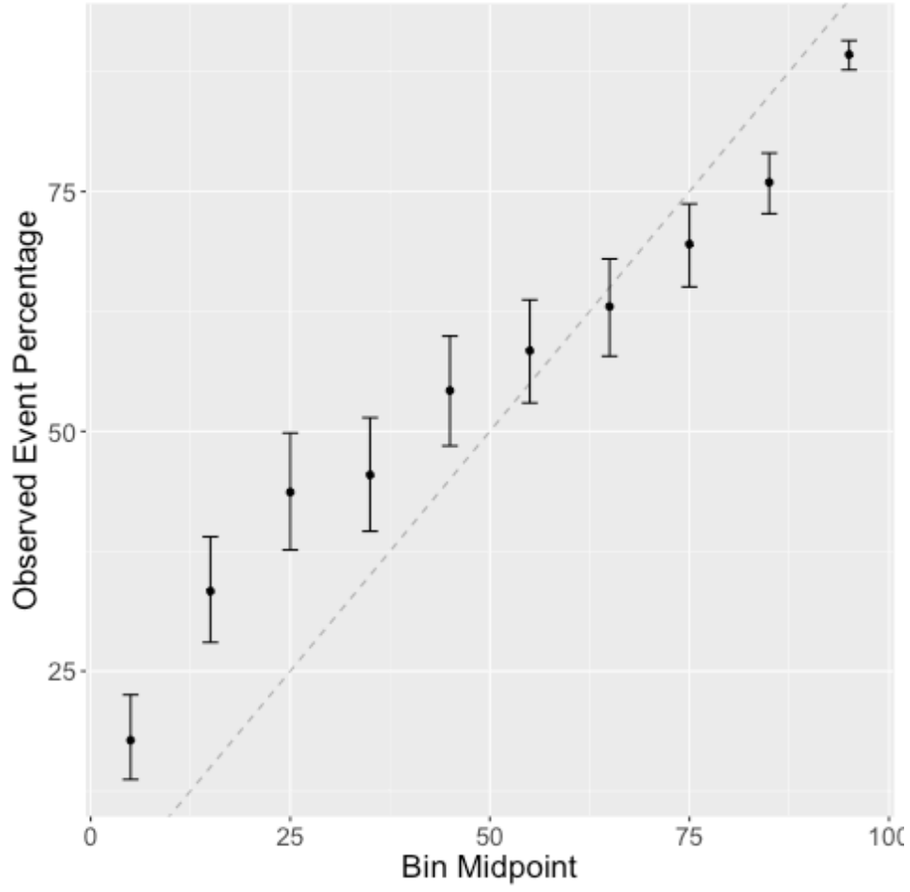


Calibration level 1: Calibration-in-the-large

- **Definition:** The observed proportion of events is equal to the average predicted probability

$$\Pr(Y = 1) = \mathbb{E}[\hat{p}(X)]$$



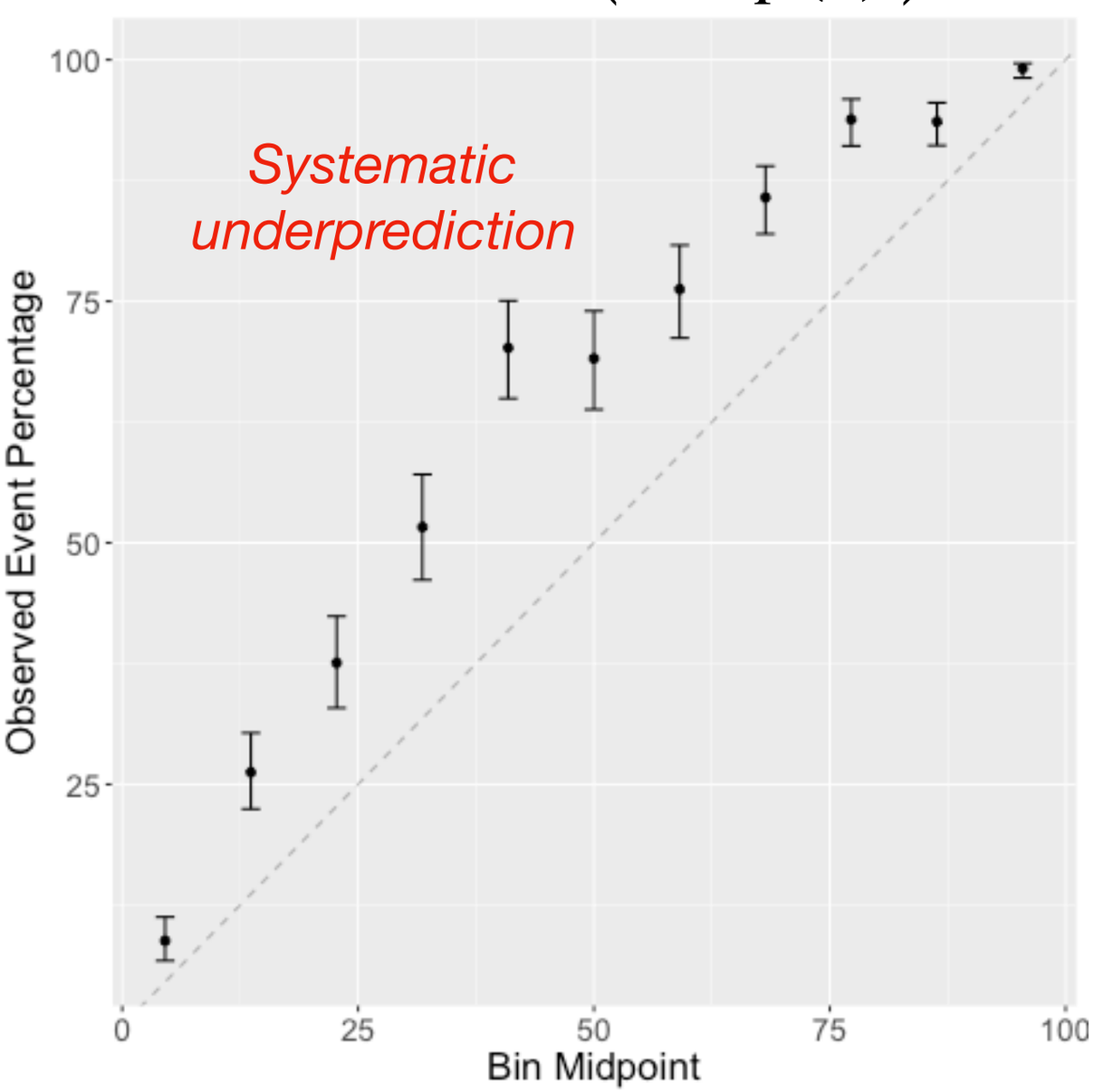


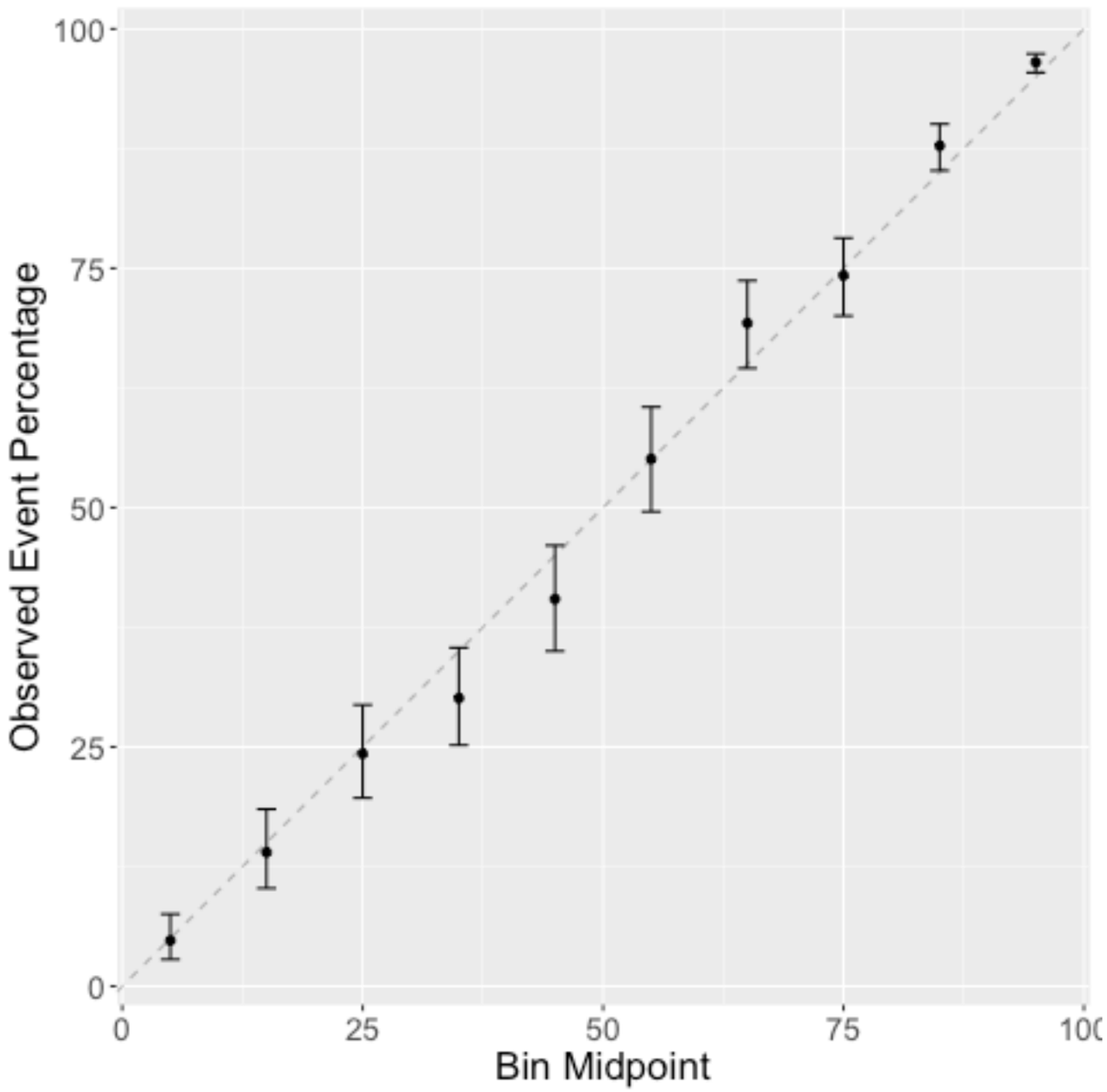
Updating for calibration-in-the-large

$$\hat{f}_{orig}(x) = \log\left(\frac{\hat{p}(x)}{1-\hat{p}(x)}\right)$$

Update the baseline risk

$$\hat{f}_{new}(x) = \hat{f}_{orig}(x) + \hat{\beta}_0$$



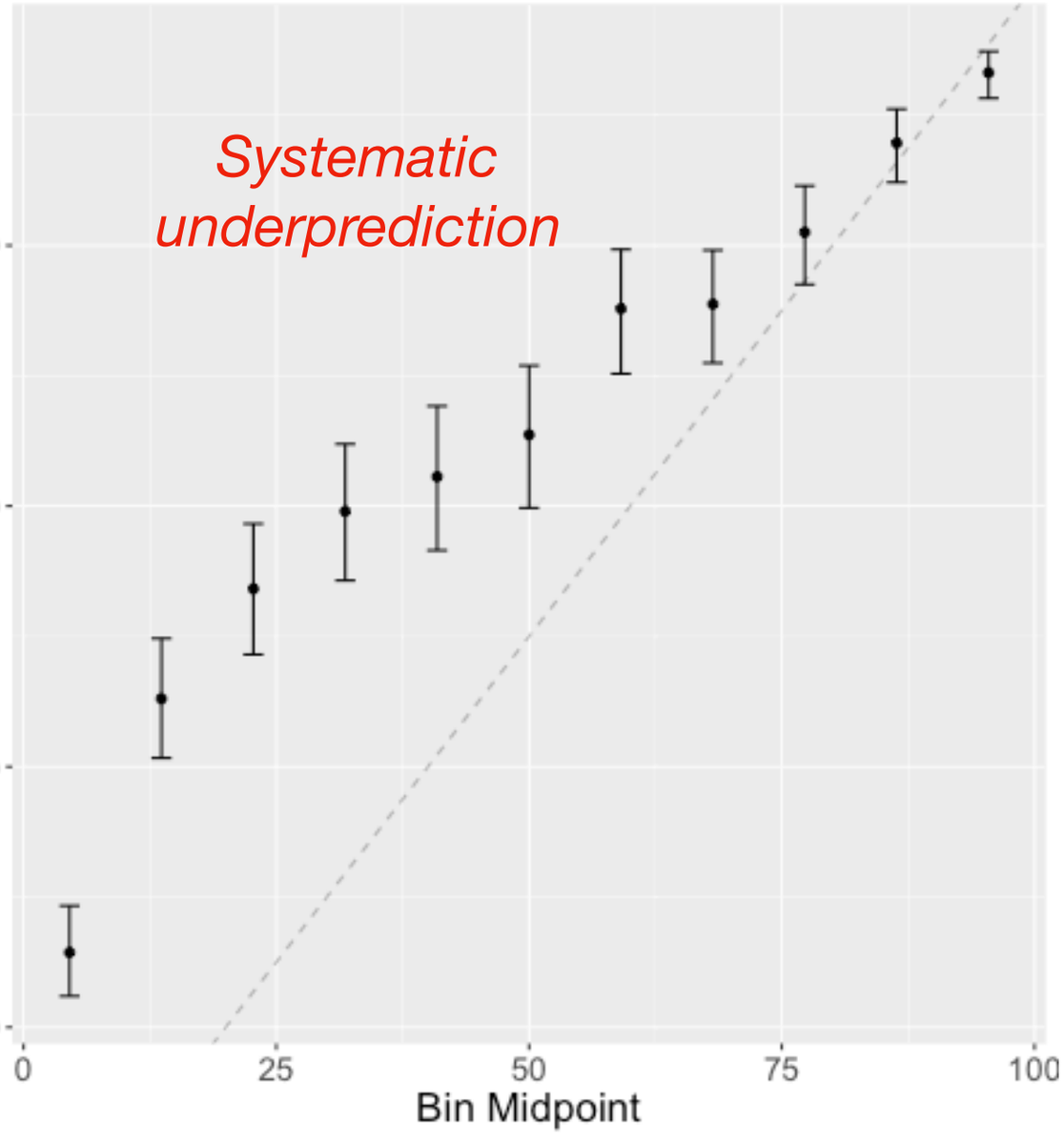


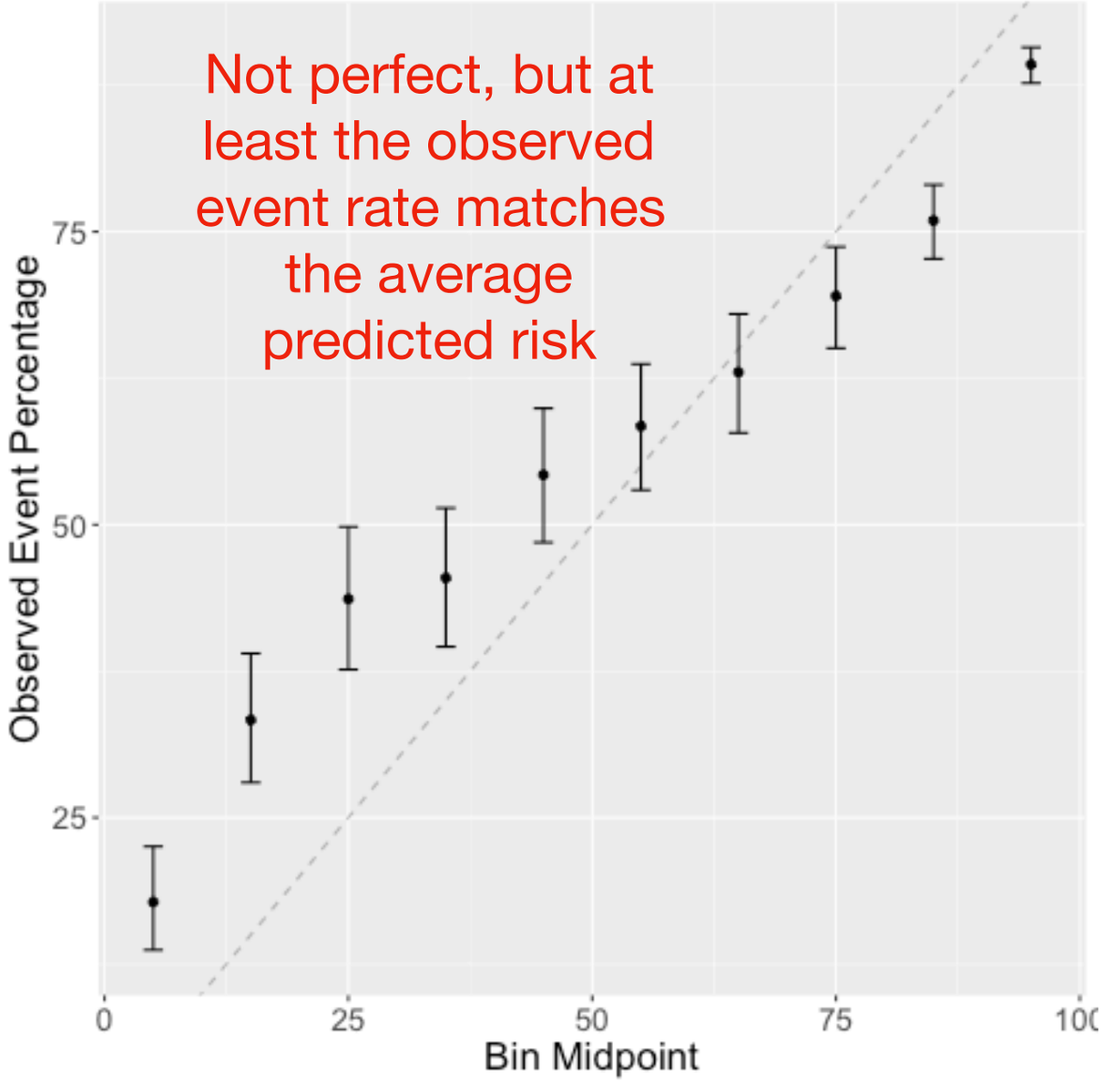
Updating for calibration-in-the-large

Let's do the same thing but for a slightly more difficult population:

$$\hat{f}_{orig}(x) = \log\left(\frac{\hat{p}(x)}{1-\hat{p}(x)}\right)$$

$$\hat{f}_{new}(x) = \hat{f}_{orig}(x) + \hat{\beta}_0$$





What type of problem is model recalibration?

- **A**: Supervised learning
- **B**: Unsupervised learning
- **C**: Semi-supervised learning

Steps for estimating a recalibration intercept

1. Create a recalibration dataset given observations (X, Y):
    1. Univariate predictor as the predicted log odds from the underlying model, e.g. $Z = \hat{f}(X)$
    2. Outcome: Y
2. Fit a logistic model on the recalibration dataset of the form $p_{new}(z) = \dfrac{1}{1 + \exp(-(Z + \beta_0))}$, where the slope of the logistic model is fixed at 1.

Code for recalibration in-the-large

To fit a logistic regression model but only estimate the intercept term, fit an intercept-only model with the “offset” set to the predicted log odds from the underlying model.

```
recalib_large_dat <- data.frame(
  pred_logit = predict(glm_res, test_dat3),
  y=test_dat3$y)
glm_large <- glm(y ~ 1, data=recalib_large_dat, family = binomial, offset=pred_logit)
```

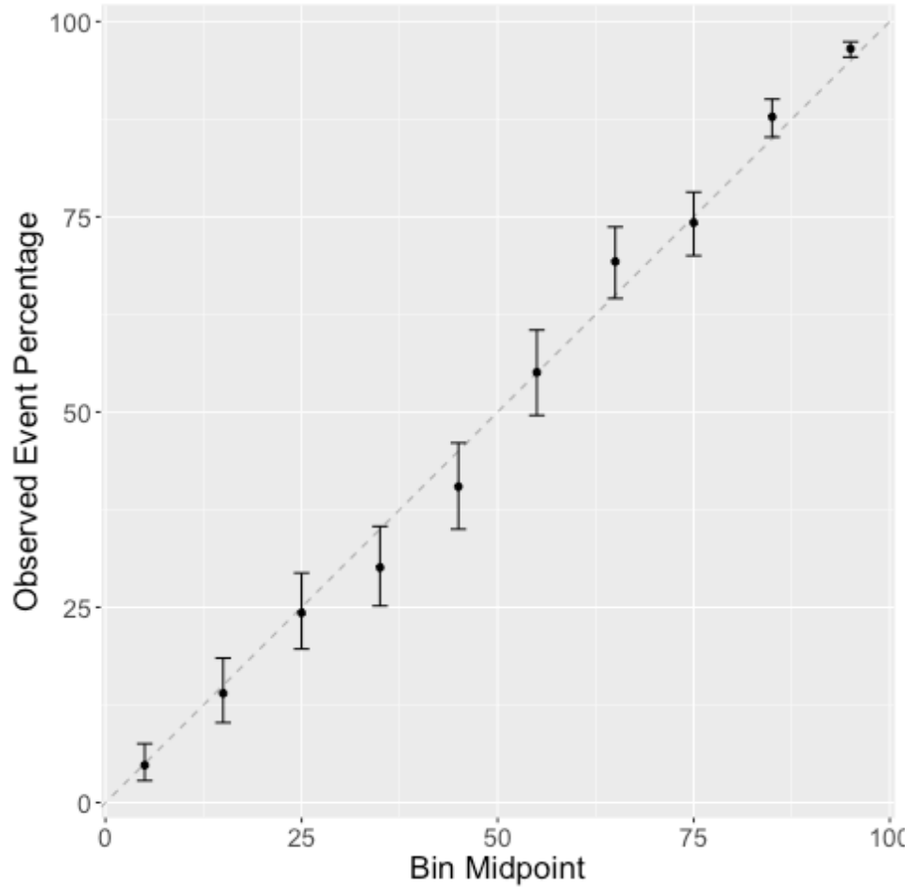
```
Call:
glm(formula = y ~ 1, family = binomial, data = recalib_large_dat,
    offset = pred_logit)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.9909  -0.6289   0.2712   0.6354   2.6518

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.90861    0.03851   23.59   <2e-16 ***
```

Calibration level 2: Weak calibration

- **Definition**: No systematic overestimation or underestimation of risks as detected by logistic regression or a Cox recalibration test.



- Fit a logistic regression model $y \sim \beta_1 \hat{f}(x) + \beta_0$, where the predicted score from the model $\hat{f}$ is treated as a variable.

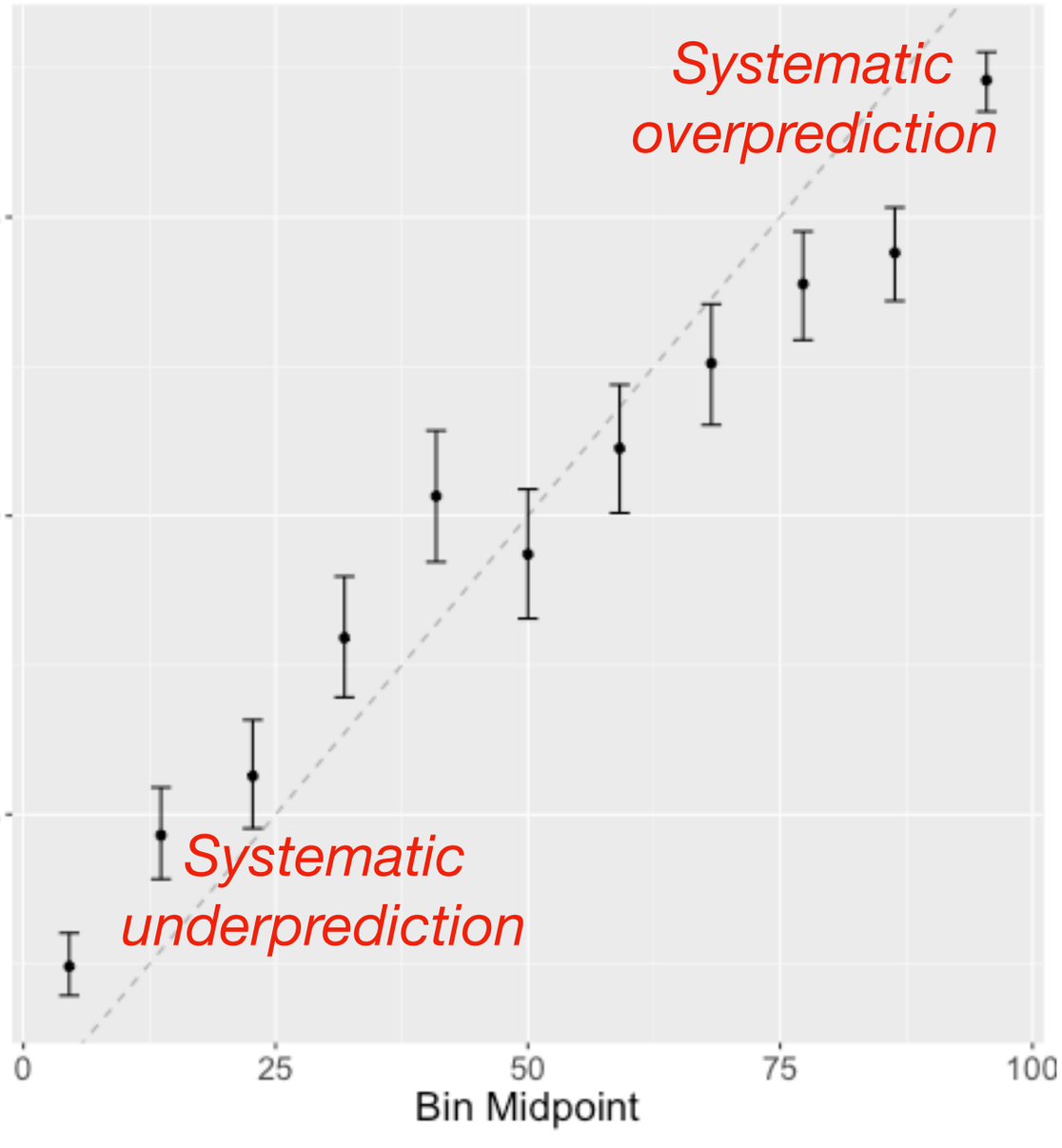
- Test the null hypothesis: $H_0 : \beta_0 = 0, \beta_1 = 1$

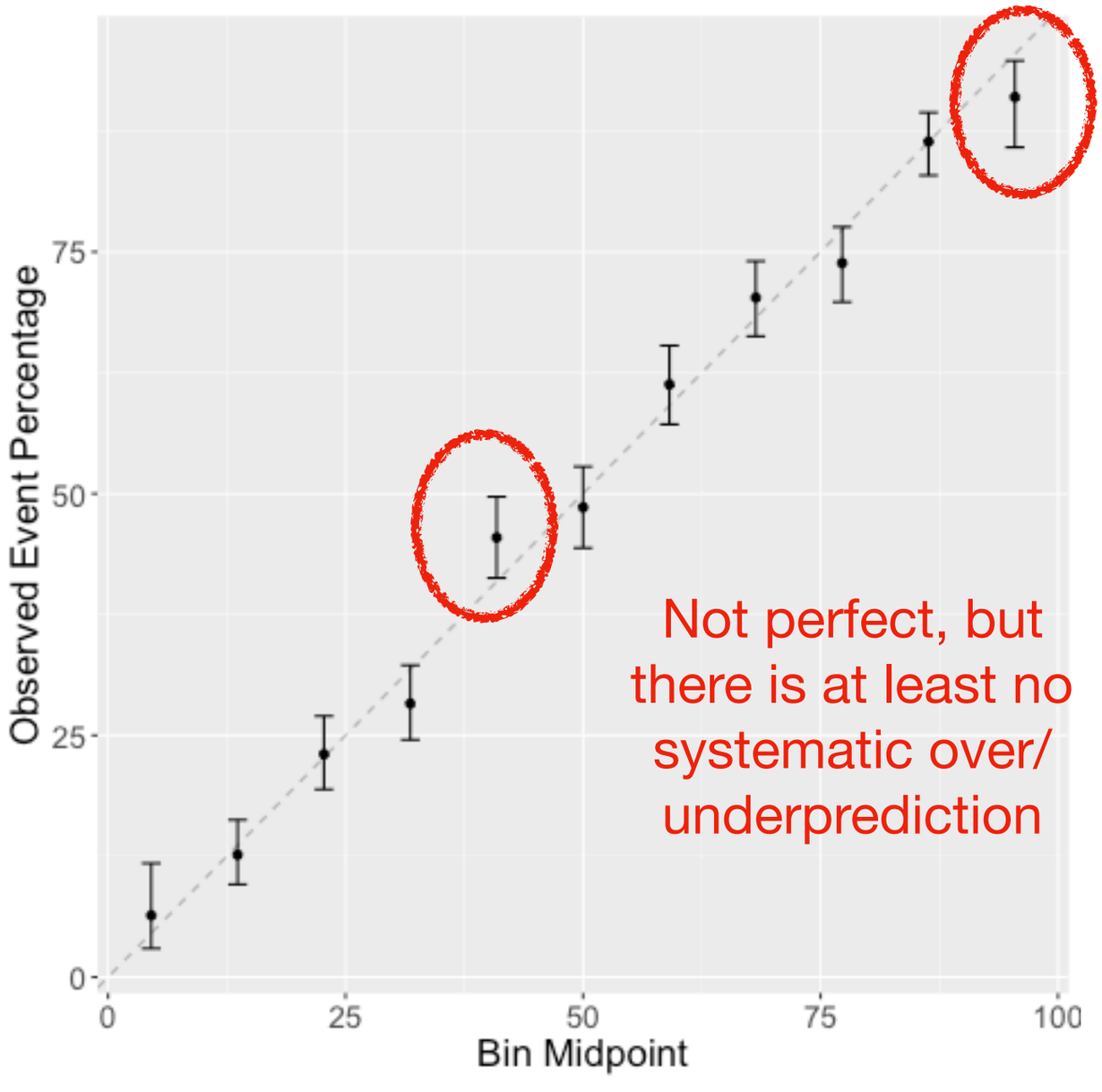
Updating for weak calibration

$$\hat{f}_{orig}(x) = \log\left(\frac{\hat{p}(x)}{1-\hat{p}(x)}\right)$$

Scale and shift the predicted log odds

$$\hat{f}_{new}(x) = \hat{\beta}_1 \hat{f}_{orig}(x) + \hat{\beta}_0$$





Steps for estimating the recalibration intercept and slope

1. Create a recalibration dataset given observations (X, Y):
    1. Univariate predictor as the predicted log odds from the underlying model, e.g. $Z = \hat{f}(X)$
    2. Outcome: Y
2. Fit a logistic model on the recalibration dataset of the form $p_{new}(z) = \dfrac{1}{1 + \exp(-(\beta_1 Z + \beta_0))}$.

Code: Estimating the calibration slope and intercept

```
recalib_weak_dat <- data.frame(
  pred_logit = predict(glm_res, test_dat2),
  y=test_dat2$y)
glm_weak <- glm(y ~ pred_logit, data=recalib_weak_dat, family = binomial)
```

```
Call:
glm(formula = y ~ pred_logit, family = binomial, data = recalib_weak_dat)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5294  -0.9275   0.2949   0.9369   2.6686

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.02467    0.03245   -0.76    0.447
pred_logit   0.59478    0.01959   30.37   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

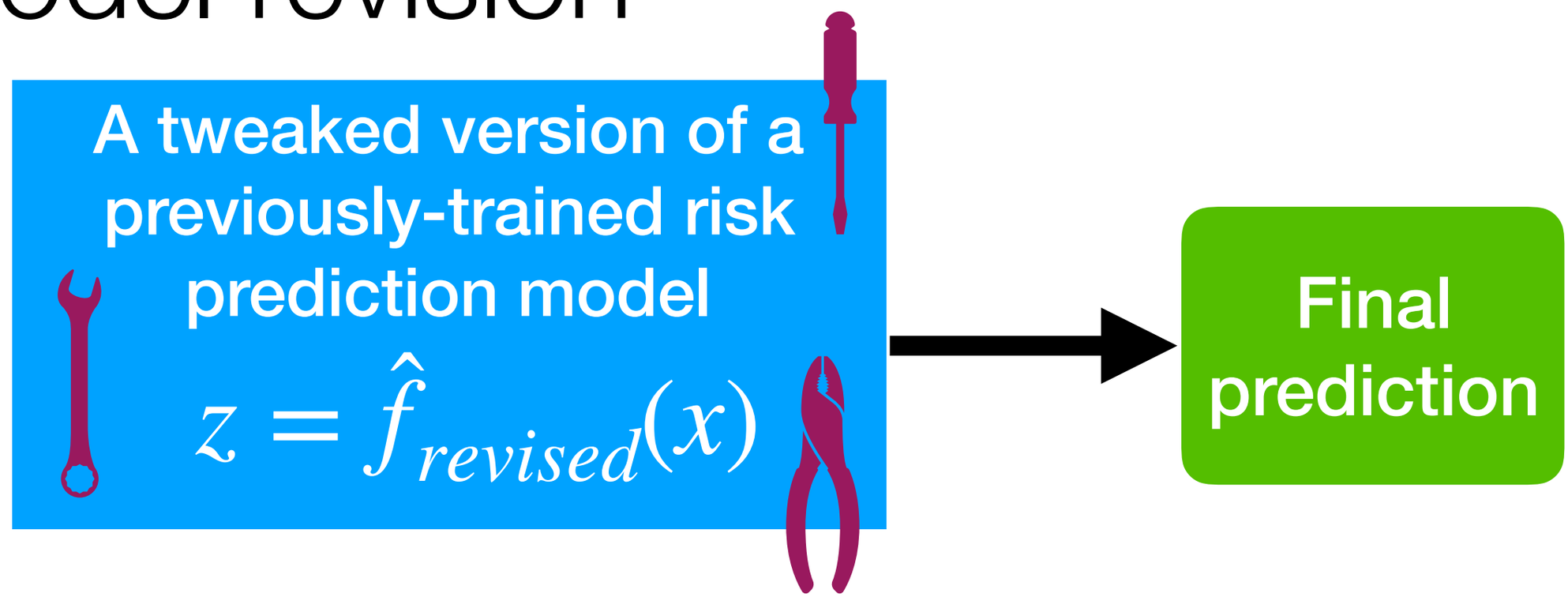
Summary

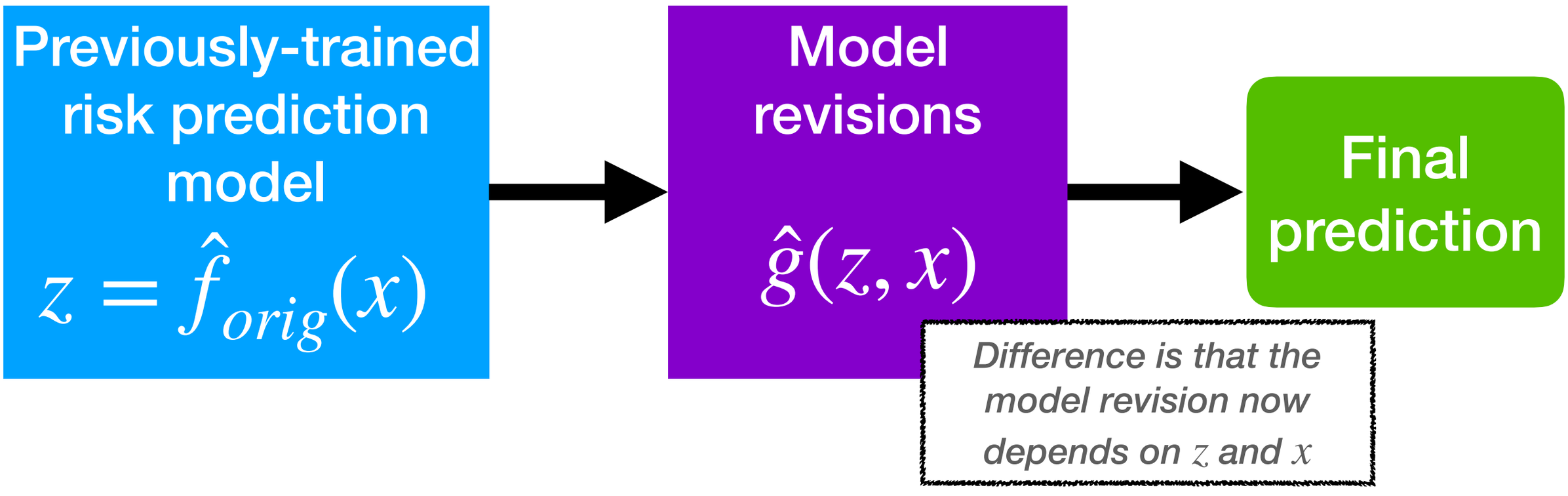
- Model recalibration procedures make “lightweight” modifications to the predicted risks from an existing prediction model.
    - To achieve calibration-in-the-large, we fit an intercept-only model.
    - To achieve weak calibration, we estimate the slope and intercept of a logistic regression model.
- The methods we presented are agnostic to the underlying model, so are suitable for black-box models.

Today's lecture

1. Motivation
2. Distributional shifts
3. Model recalibration
4. Model revision
5. Selecting the best update
6. Model-specific update methods
7. Online model updating

Model revision





Model revision

- To protect against distributional shifts, one may need more extensive model updating procedures.

- Model recalibration methods focus on shifting and scaling the predicted risks. These methods adjust for an average shift across all variables.

- Sometimes shifts have occurred only in a *subset* of the variables $S \subseteq \{1,...,p\}$. In this case, we may want to consider revising the model for only that subset.

Model revision for a subset of variables

- Consider again our COPD example. How can we revise the model just for a subset of variables, say X2 and X4?

Original model

	Estimate
(Intercept)	-0.05966
X1	1.00883
X2	0.92354
X3	1.15324
X4	1.02586
X5	1.11122

New model

	Estimate
(Intercept)	-0.05966
X1	1.00883
X2	?
X3	1.15324
X4	?
X5	1.11122

Model revision for a subset of variables

- If we want to revise the coefficients for a subset of variables $S \subseteq \{1,...,p\}$, we can write the model revision as:

$$\hat{f}_{new}(x) = \beta_0 + \beta_1 \hat{f}_{orig}(x) + \sum_{s \in S} \beta_{2,s} x_s.$$

- So all we need to do is to fit a logistic regression model with predictors:
  - The predicted log odds from the original model $\hat{f}_{orig}$
  - Those in subset S

Model revision for a subset of variables

```
> glm_res$coefficients  // Original model
(Intercept)           X1           X2           X3           X4           X5
 -0.0596638    1.0088288    0.9235427    1.1532388    1.0258634    1.1112178
```

```
recalib_revis_dat <- data.frame(  // Create recalibration dataset
  X2=test_dat2$X2,
  X4=test_dat2$X4,
  pred_logit = predict(glm_res, test_dat2),
  y=test_dat2$y)
glm_revis <- glm(y ~ ., data=recalib_revis_dat, family = binomial)
```

// Fit a revised model

```
Call:
glm(formula = y ~ ., family = binomial, data = recalib_revis_dat)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2980  -0.7632  -0.2379   0.7922   2.1466

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.1716     0.1822   0.942    0.346
X2            0.2308     0.2050   1.126    0.260
X4           -0.9429     0.2218  -4.251 2.13e-05 ***
pred_logit    0.8902     0.1403   6.345 2.23e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

	Estimate
(Intercept)	-0.05966
X1	1.00883
X2	***1.154***
X3	1.15324
X4	***0.083***
X5	1.11122

Model revision for a subset of variables

- Note: We can use model revisions of the form $\hat{f}_{new}(x) = \beta_0 + \beta_1 \hat{f}_{orig}(x) + \sum_{s \in S} \beta_{2,s} x_s$ to revise *any* risk prediction model, even black-box models. This revision only depends on the output of the original model.

Model revision using shrinkage methods

- What if we don't know which coefficients need updating?
- In this case, we could use **best variable subset selection methods** or **shrinkage methods** like the ridge or the Lasso.
- Example using the Lasso for model revision:

$$\min_{\beta} \frac{1}{n} \sum_{i=1}^{n} \ell \left(y_i, \hat{f}_{new}(x) \right) + \lambda \|\beta\|_1$$

where $\hat{f}_{new}(x) = \beta_0 + \hat{f}_{orig}(x) + \beta^\top x.$

This tries to shrink the change in model coefficients towards zero. In other words, we are encouraging the new model coefficients to be close to the original model coefficients.

Today's lecture

1. Motivation
2. Distributional shifts
3. Model recalibration
4. Model revision
5. Selecting the best update
6. Model-specific update methods
7. Online model updating

Selecting the most appropriate update

- We just covered many ways to update a model. How do we pick the best one for our dataset?

- **A**: Training validation split

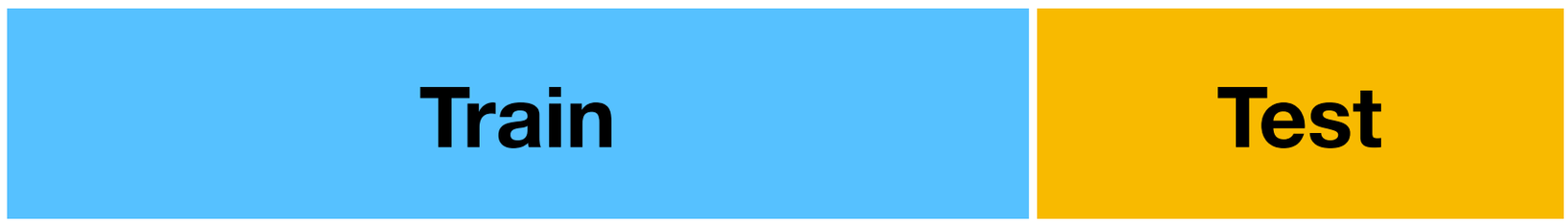
- **B**: Cross validation

- **C**: Pick the simplest model update that isn't too different from the update with the highest accuracy.

- **D**: Hypothesis testing

Training-validation split for selecting a model update

Shuffled dataset: 7 22 13 15 56 38 …



Candidate methods

1. Adjust baseline risk
2. Scale and shift predicted risks
3. Revise the associations for a subset of the variables

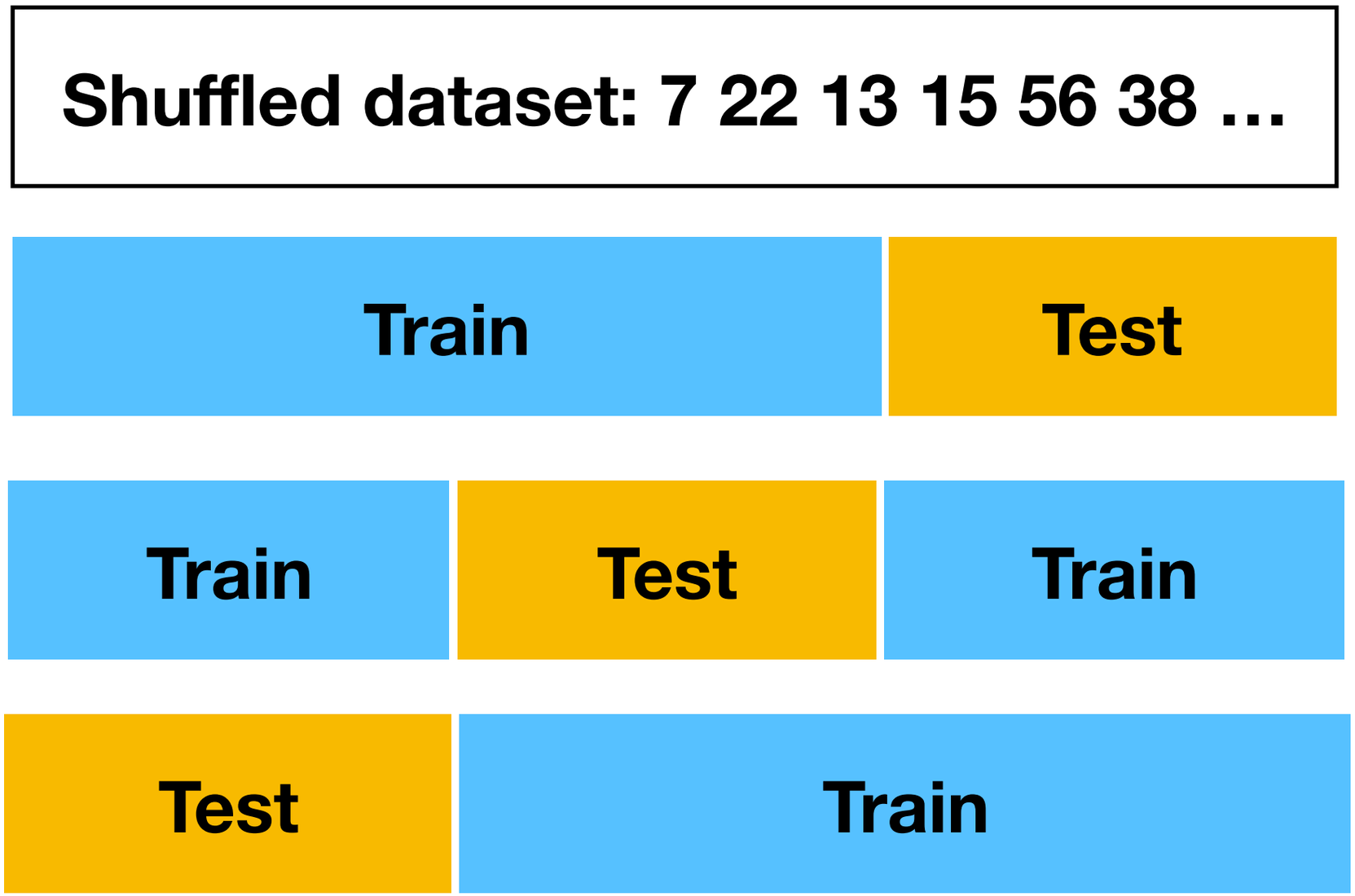
…

Test errors

1. 0.9
2. 0.8
3. 0.85

…

Cross-validation for selecting the model update



Selecting the simplest update that performs almost as good as the best

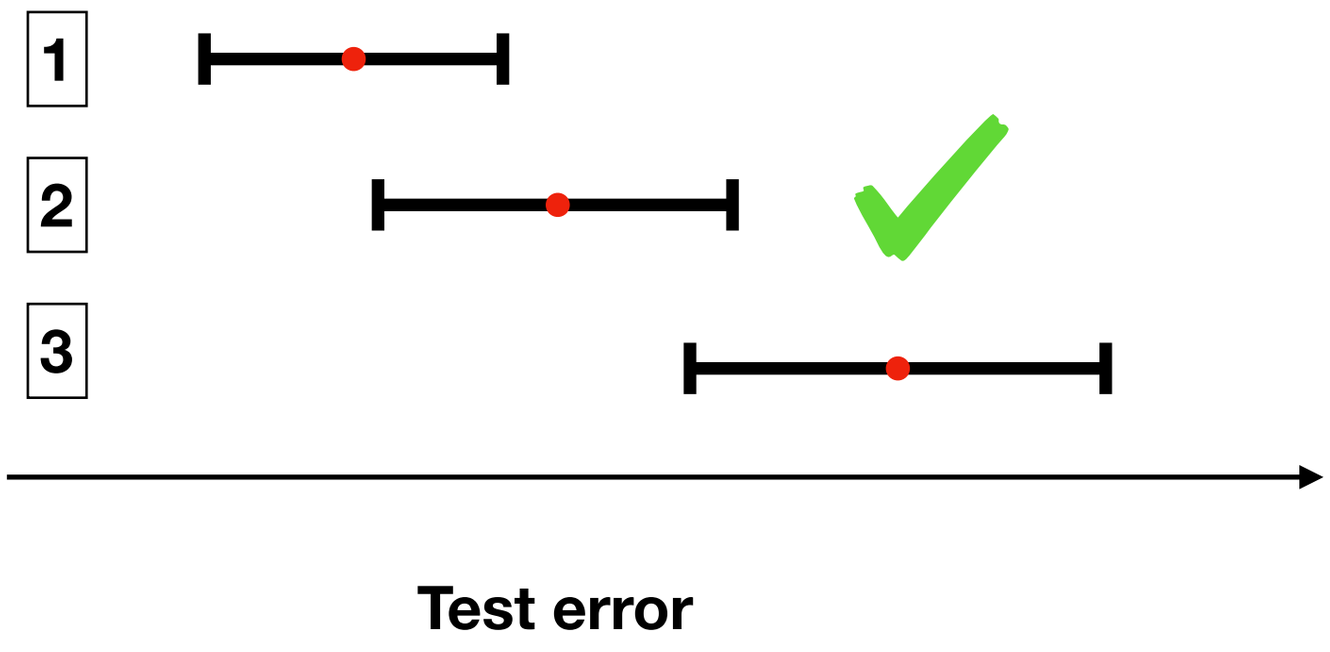
- Construct $(1-\alpha)$-confidence intervals for the expected test loss for each model update procedure.

- Pick the method that is within some acceptable margin, say one standard error, of the best performing method.

Candidate methods

1. Adjust baseline risk
2. Scale and shift predicted risks
3. Revise the associations for a subset of the variables

…



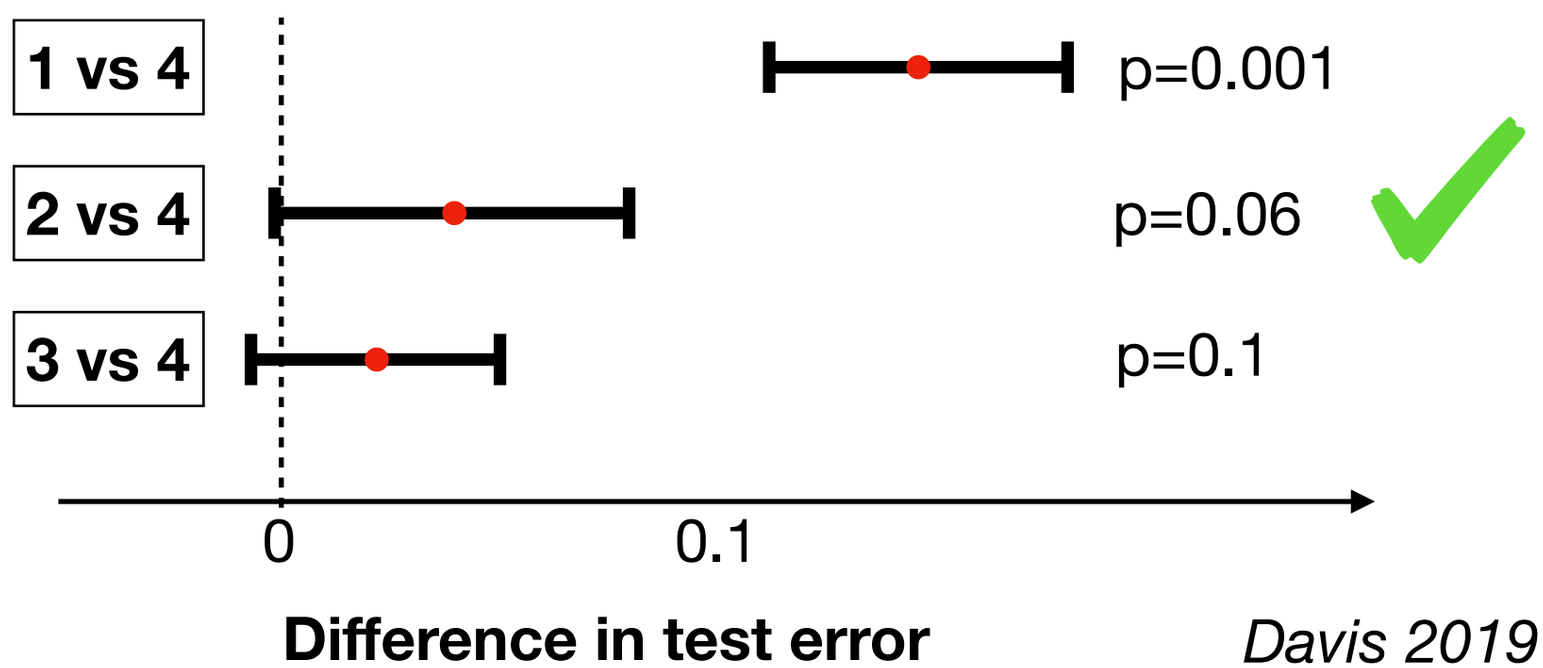
Selecting the simplest update by hypothesis testing

- Run a hypothesis tests that compares candidate model updates to the “ideal” model update, e.g. fit a new model.

- Recommend the simplest updating method whose p-value greater than some level α, i.e. choose the simplest method that was not statistically significantly worse than the “ideal” model update.

Candidate methods

1. Adjust baseline risk
2. Scale and shift predicted risks
3. Revise the associations for a subset of the variables
4. Fit a new model



Davis 2019

Today's lecture

1. Motivation
2. Distributional shifts
3. Model recalibration
4. Model revision
5. Selecting the best update
6. Model-specific update methods
7. Online model updating

Model-specific update methods

- The methods we've discussed so far are agnostic to the original risk prediction model.

- There are also model-specific updates that can be useful.

Updating a CART model

- A radical revision of a CART model is to fit an entirely new tree.

- A simpler strategy is to keep the tree structure fixed and to re-estimate the mean/mode at each leaf.

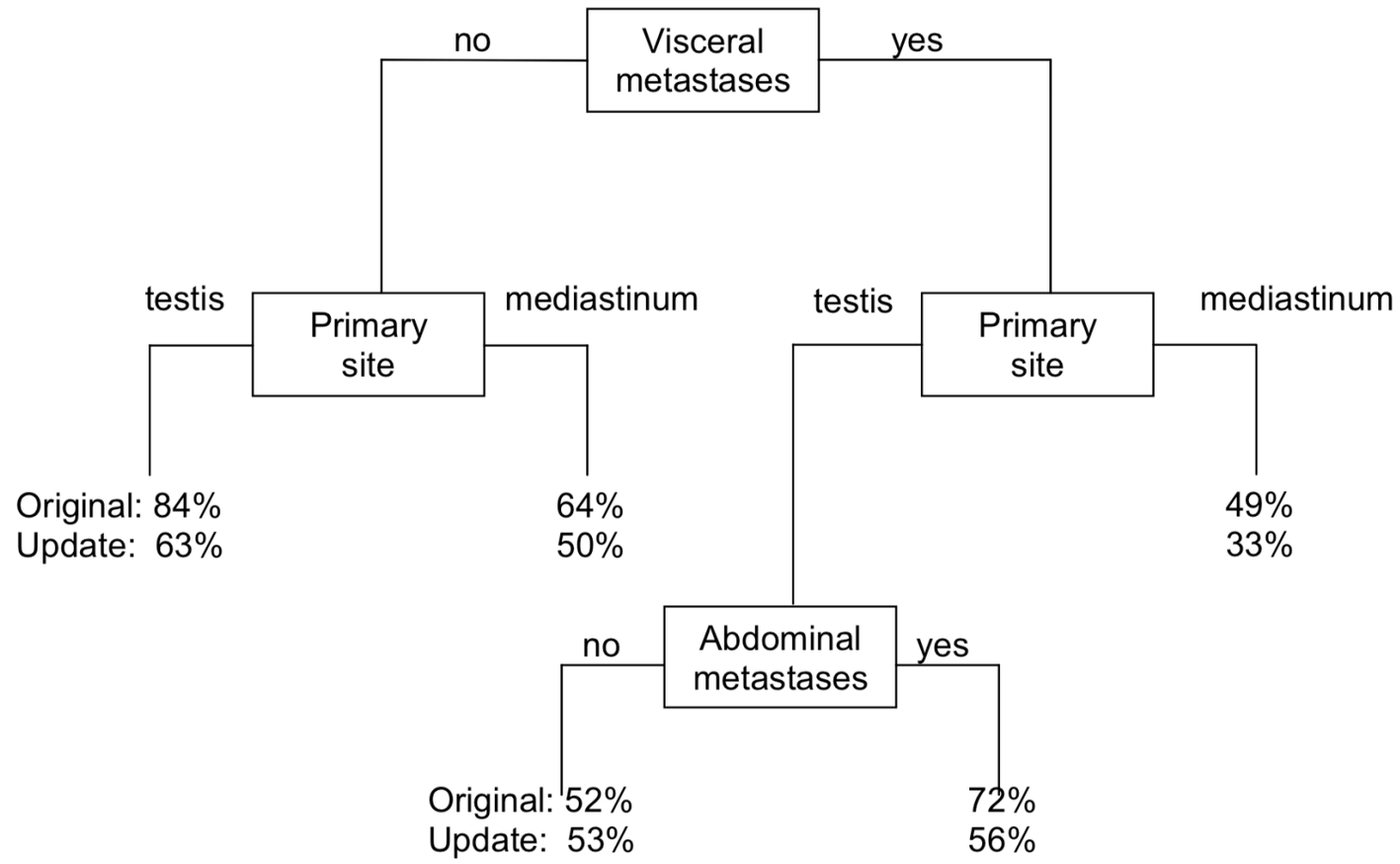


Fig. 20.6 Regression tree as developed for poor prognosis patients with non-seminomatous germ cell cancer. The 2-year survival is shown for the development sample ("Original", $n = 332$) and for the validation sample ("Update", $n = 456$) [616]

Steyerberg 2019

Q: Which updating method requires more data?

- **A:** Perform model recalibration. Scale and shift the predicted score from the original CART model.

- **B:** Re-estimate the value at each leaf in the CART model.

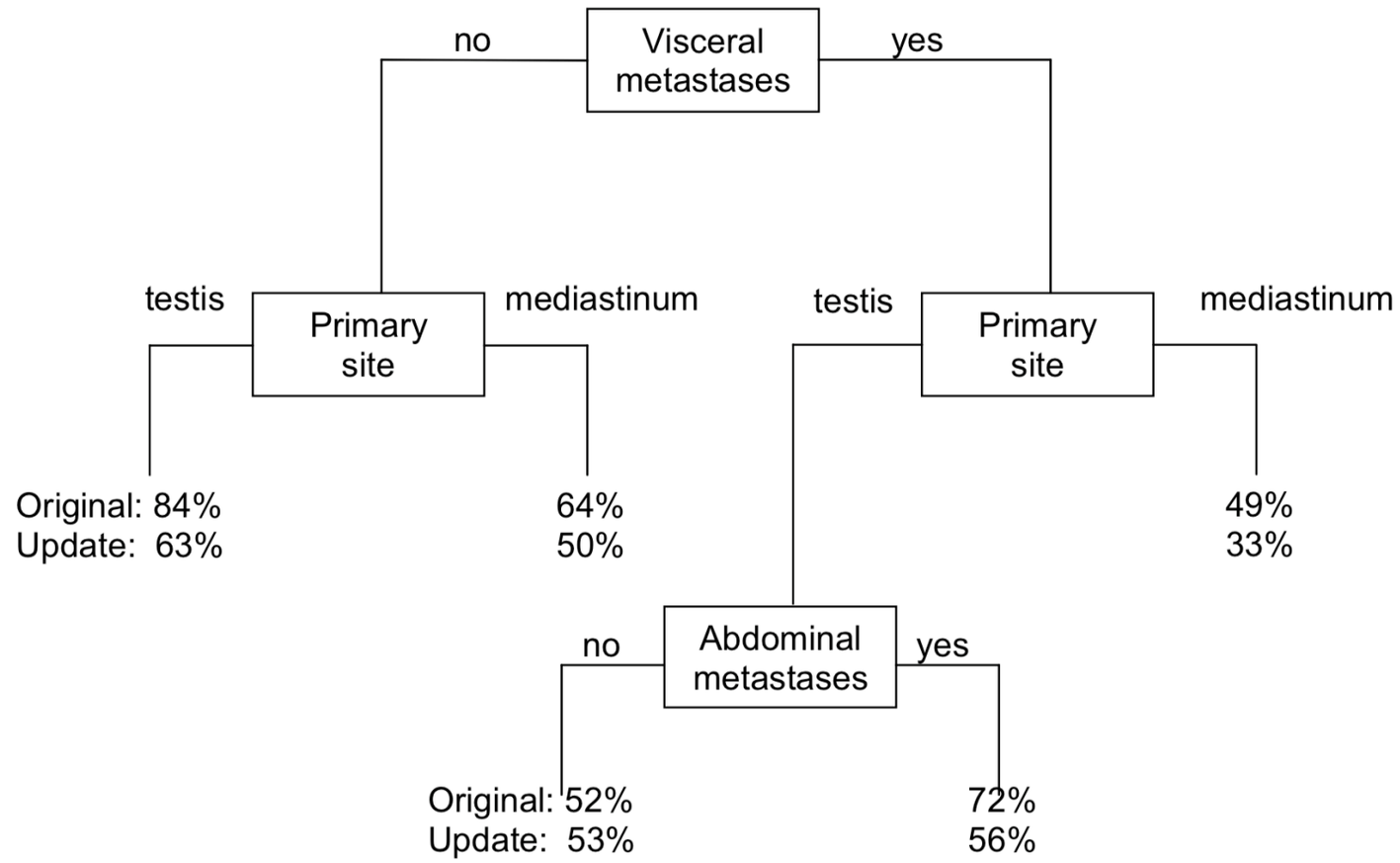
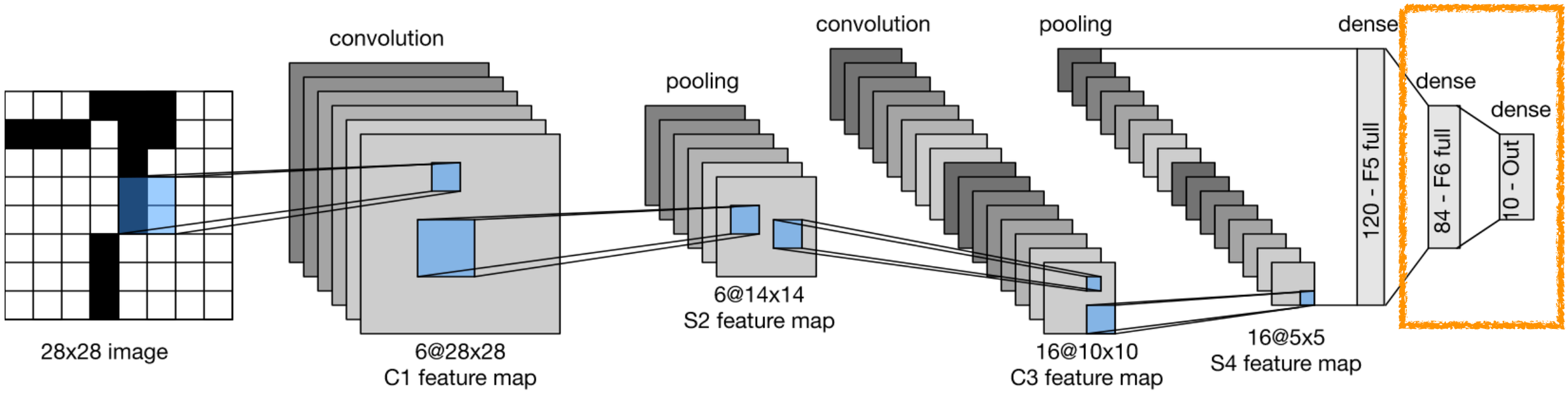


Fig. 20.6 Regression tree as developed for poor prognosis patients with non-seminomatous germ cell cancer. The 2-year survival is shown for the development sample ("Original", $n = 332$) and for the validation sample ("Update", $n = 456$) [616]

Steyerberg 2019

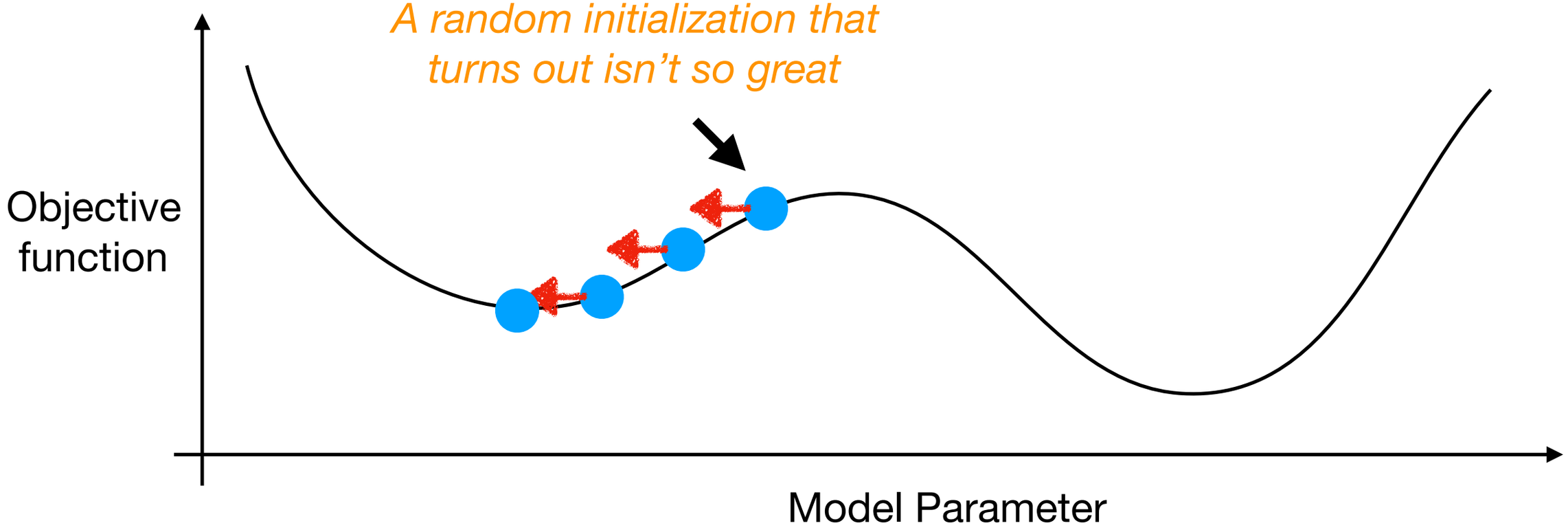
Updating a neural network

- If you think the features extracted from a neural network would be helpful but you don't have enough data (or time/ energy) to retrain the entire model, one solution is to lock the parameters of the neural network from the early levels and only retrain the last few levels.
- For example, we may choose to retrain only the last layer.



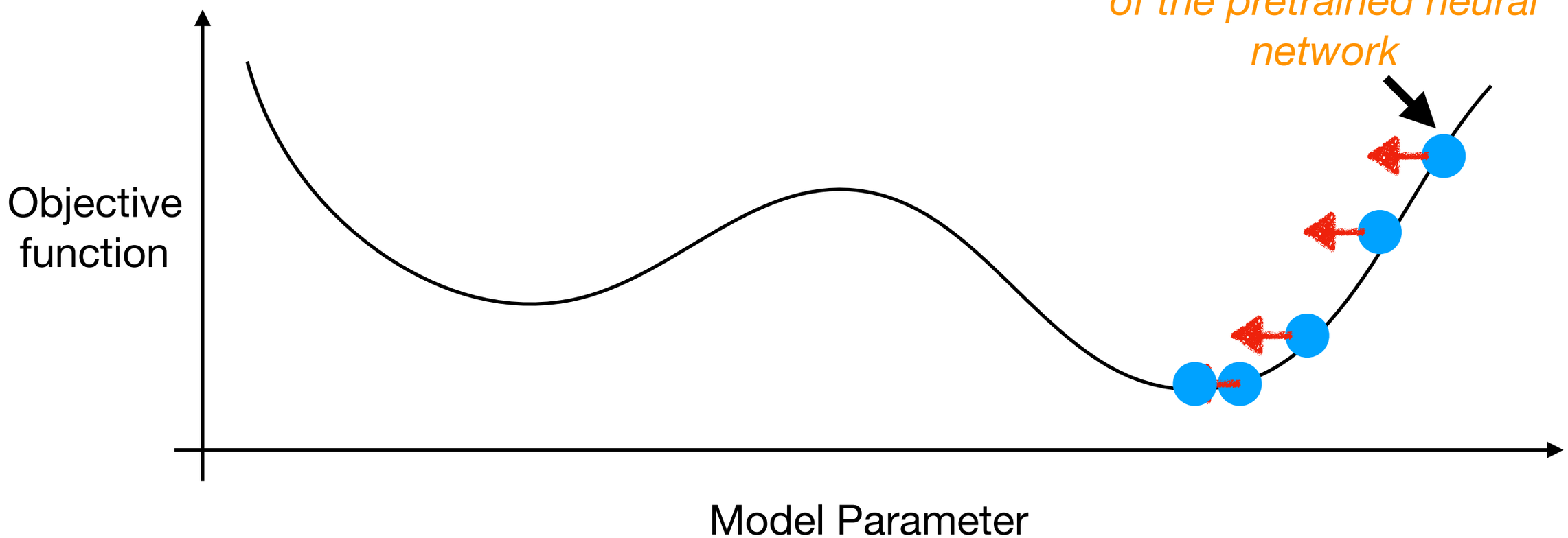
Updating a neural network

- If you do have a lot of data, you could retrain the entire neural network as well, but initialize the neural network parameters using pretrained parameters.



Updating a neural network

- If you do have a lot of data, you could retrain the entire neural network as well, but initialize the neural network parameters using pretrained parameters.



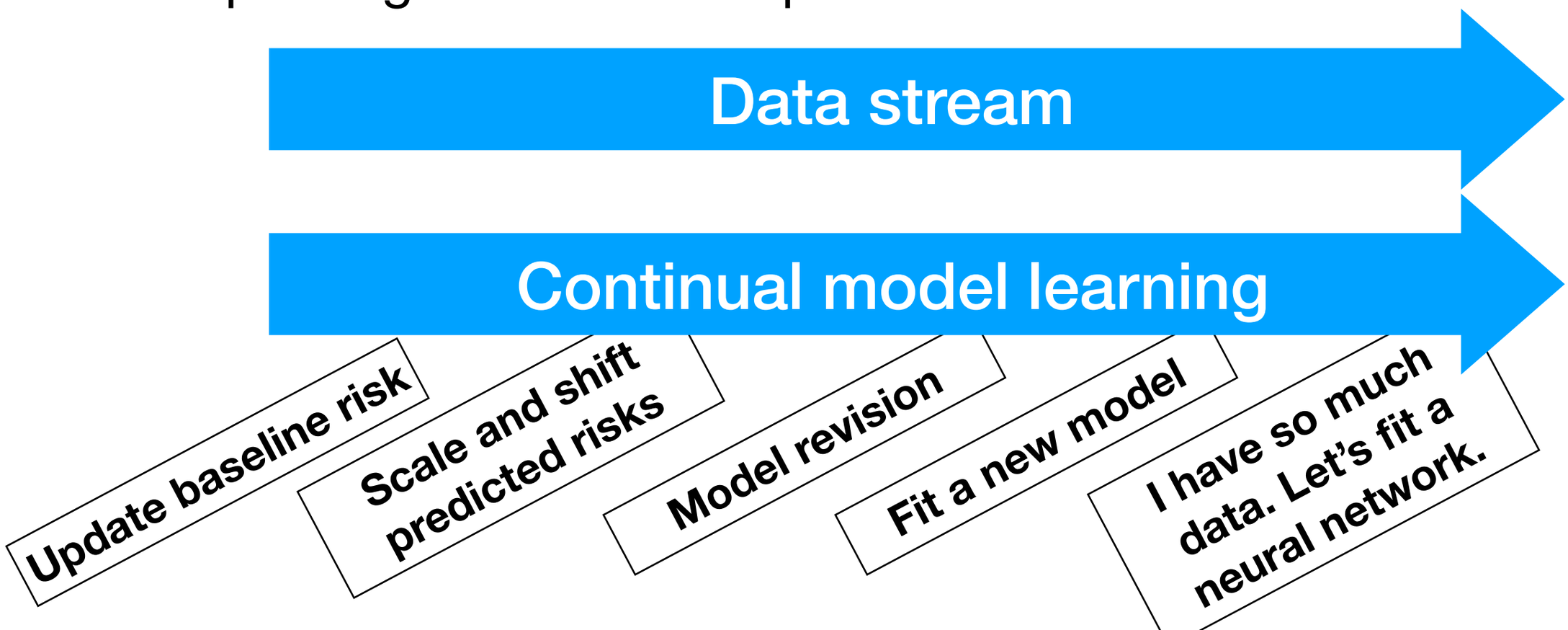
Today's lecture

1. Motivation
2. Distributional shifts
3. Model recalibration
4. Model revision
5. Selecting the best update
6. Model-specific update methods
7. Online model updating

Online model updating

- When models are deployed over long periods of time, distributional shifts are bound to occur and performance decay will likely happen.

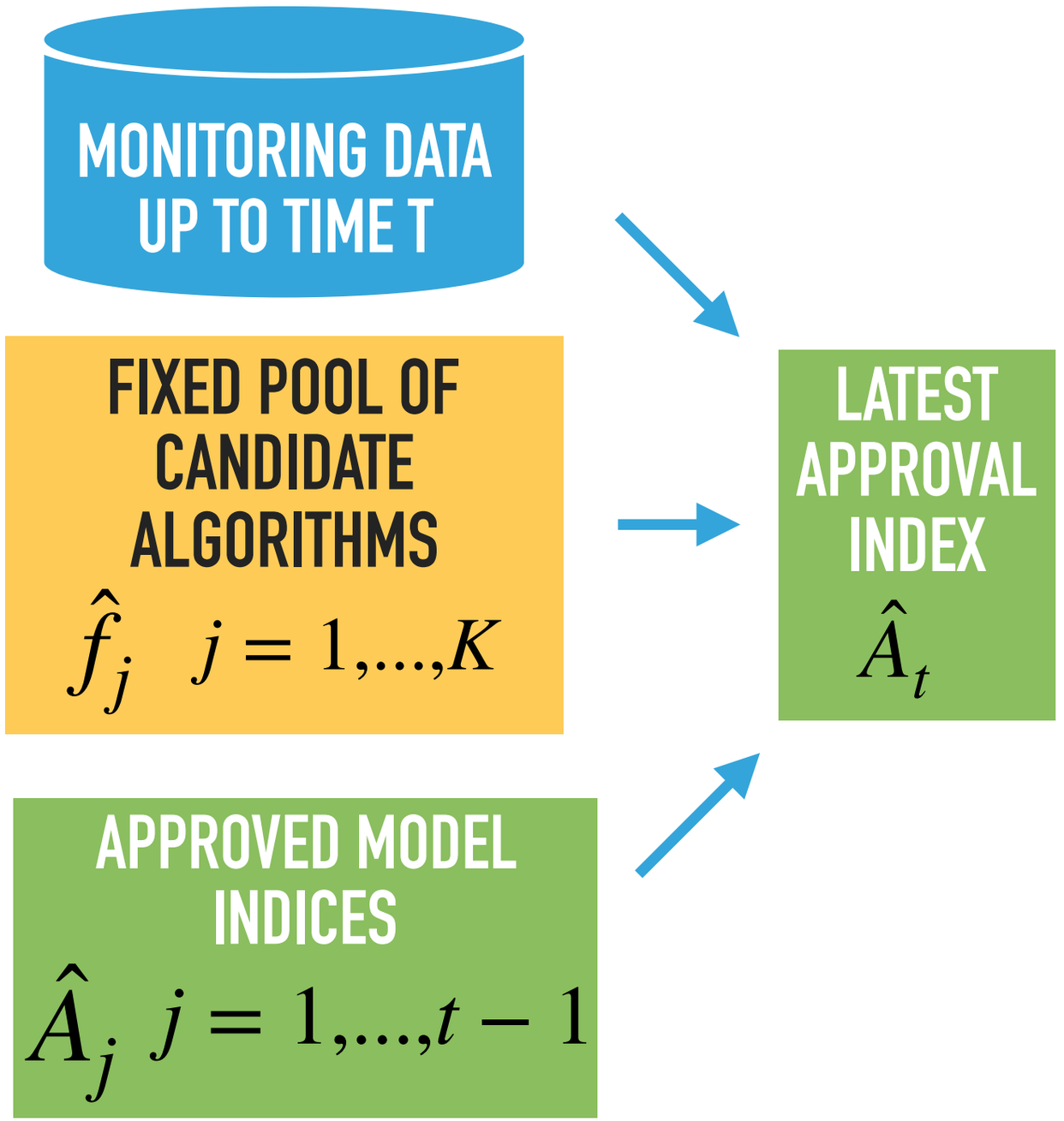
- If streaming data is available (e.g. accumulating EHR data), we may be able to do continuous monitoring and/or updating of clinical risk prediction models.

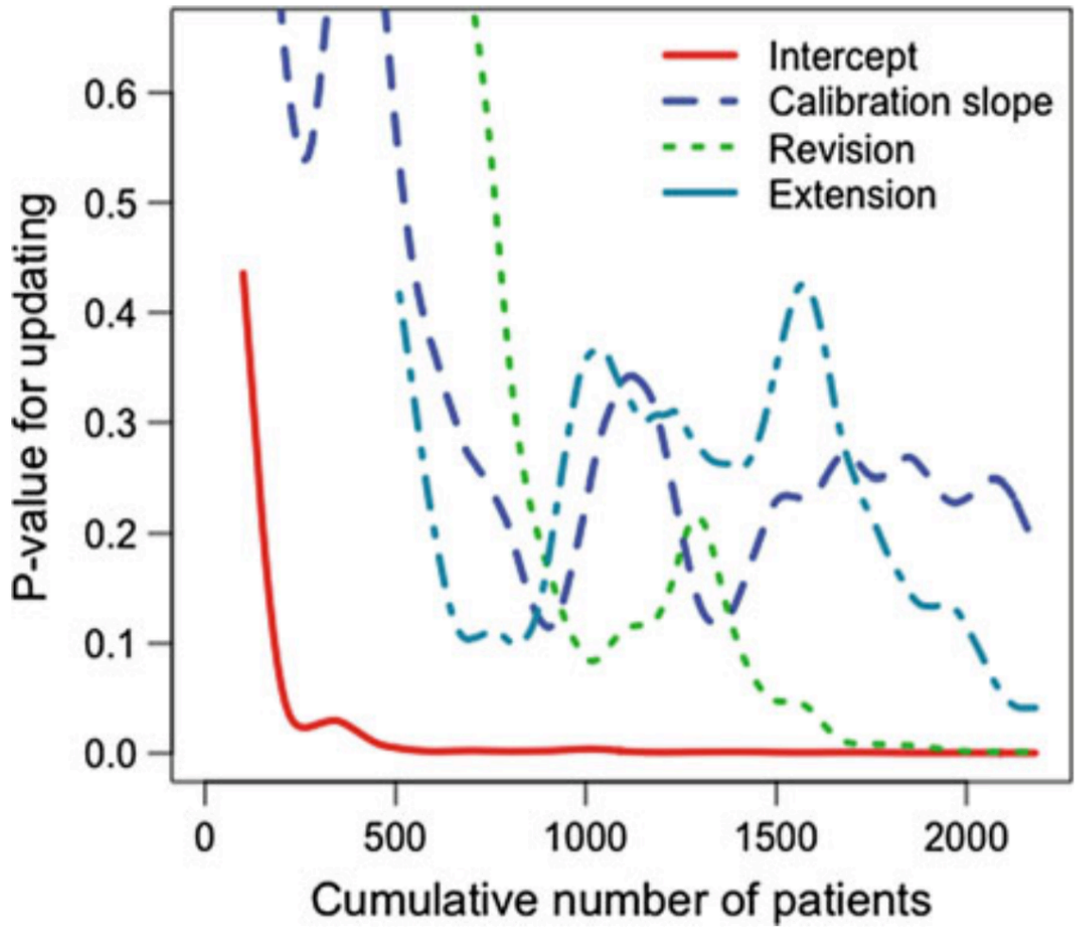


The challenge of “over-updating” in online model updating

- Online methods need to balance recency of the data with sampling noise. They need to answer questions like:
    - Is the estimated drop in performance simply a reflection of sampling noise or does it represent an actual distribution shift?
    - Using data from last week, we estimate the performance of the proposed model update to be better than the original model. Was last week’s a blip? Or is this model update actually beneficial? Or is this all just sampling noise?

Continuous p-values for model updates of different complexities

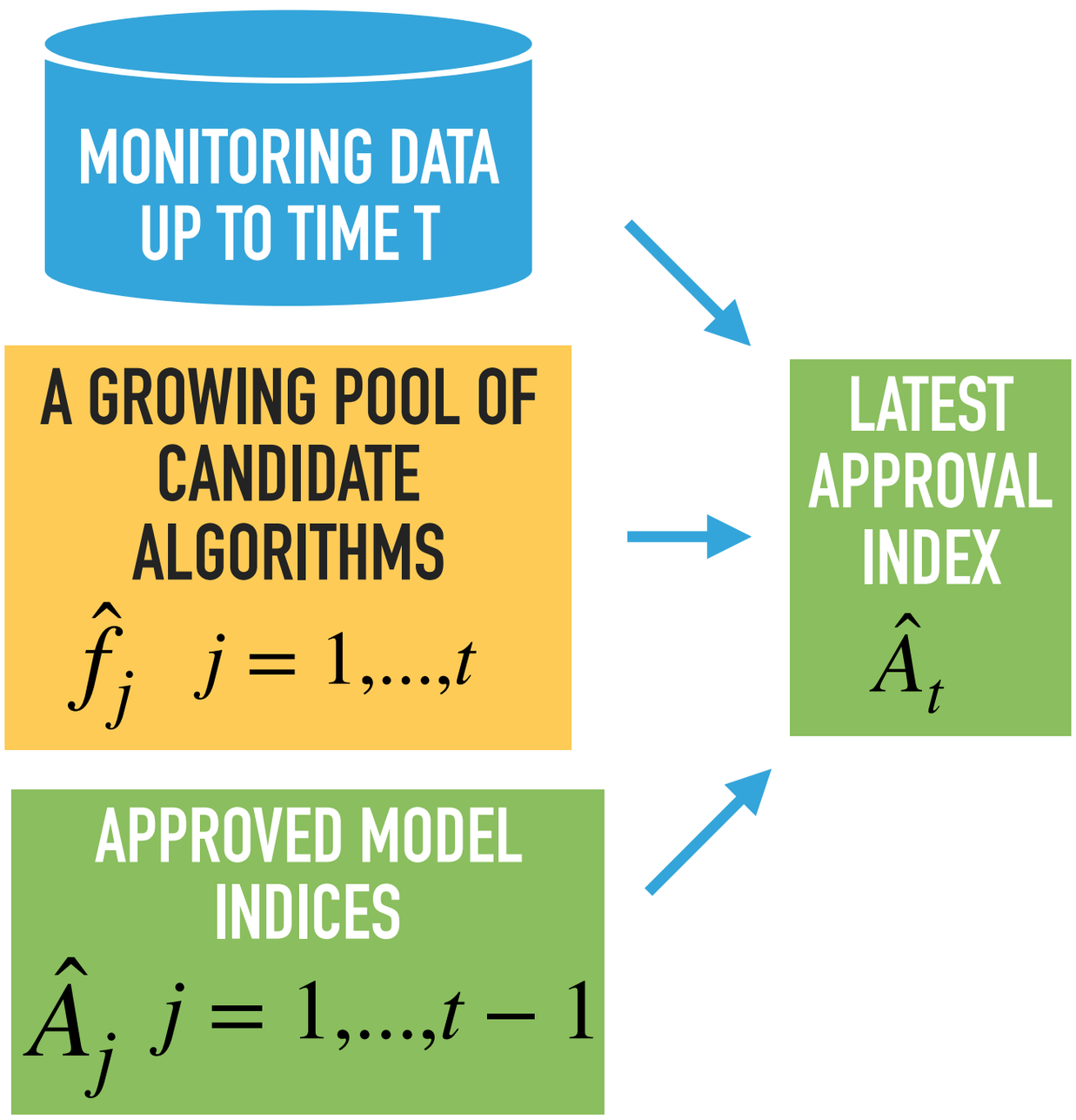




Steyerberg 2019

Validating and approving new model updates over time

- In a continually learning system, a model developer may even propose new modifications to their ML algorithm over time. In this case, we have a growing pool of candidate modifications.



Research questions:

- When constructing the hypothesis test, which algorithm do we use as the reference/baseline?

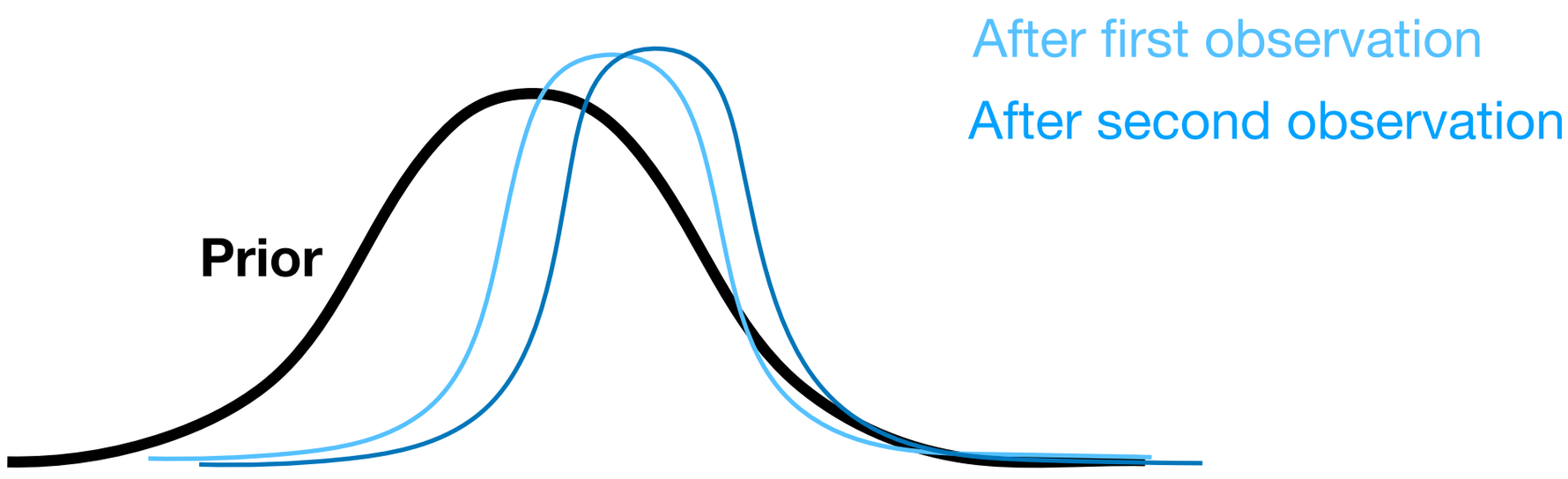
- A growing number of candidate algorithms means we have a growing number of hypothesis tests to run. How do we appropriately handle multiple hypothesis testing in online settings?

Feng 2020

Online learning using bayesian filtering

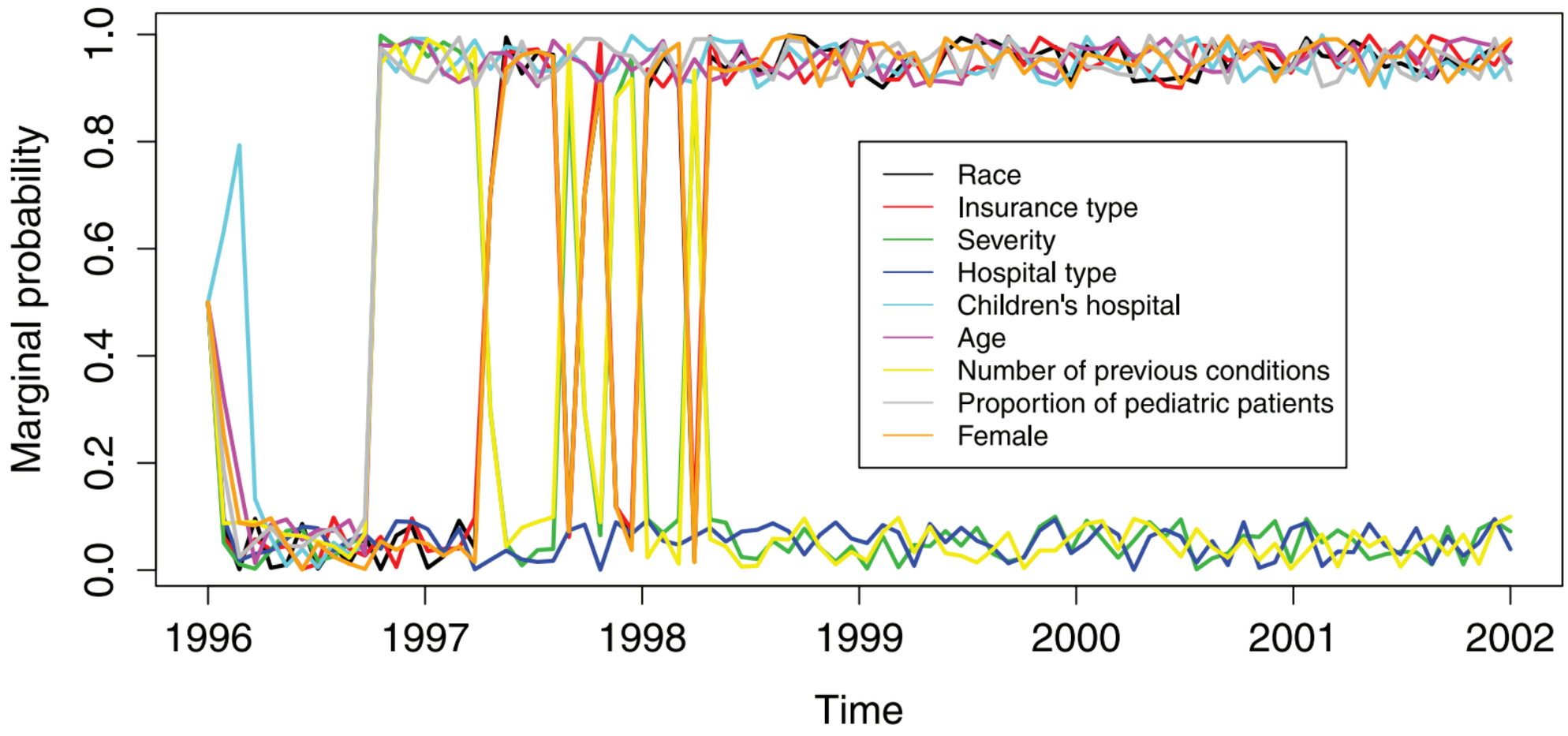
- Bayesian methods are designed for learning from accumulating data. Every time we get a new observation, we get to update our posterior.

- We can actually reformulate a lot of the model updating procedures from lecture so far using a Bayesian perspective.



Example: Dynamic logistic regression by Bayesian filtering

- Dynamic logistic regression model continually updates the posterior for the coefficients of a logistic regression model.
- Analyzing data from children with appendicitis, McCormick et. al. 2012 analyzed the factors that influenced who would receive an open appendectomy or a laparoscopic procedure.



Today's lecture

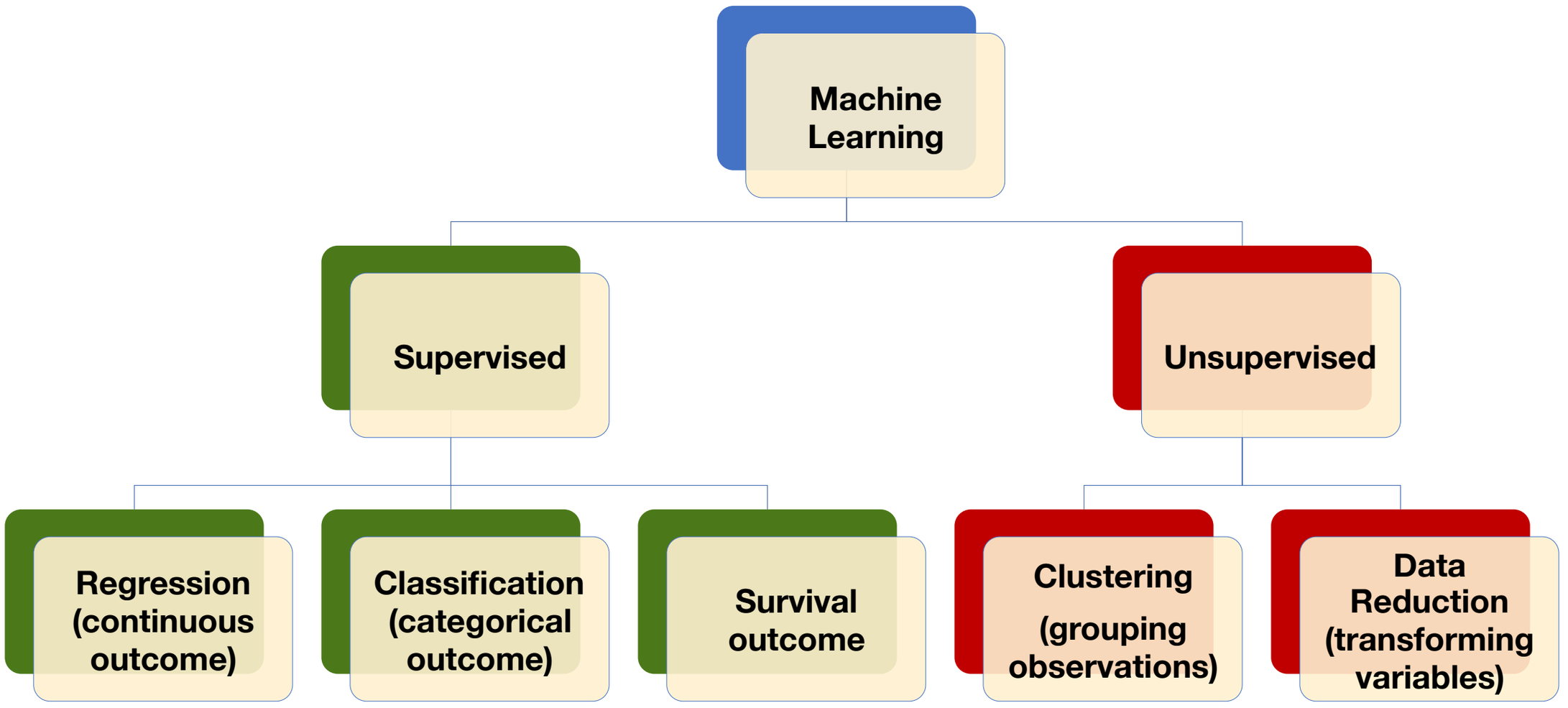
1. Motivation
2. Distributional shifts
3. Model recalibration
4. Model revision
5. Selecting the best update
6. Model-specific update methods
7. Online model updating

Additional References

- Chapter 20 of “Clinical Prediction Models” by Ewout Steyerberg

Course summary

- Going back to the course objectives from Lecture 1, I hope we've achieved the following as a class:
    - Developed intuition and a mechanistic understanding for some important machine learning methods
    - Gained insight as to which methods are most appropriate in which situations
    - Learned best practices in machine learning
    - Discovered potential pitfalls to avoid in machine learning
    - Applied machine learning methods to real datasets in R


Machine Learning
Supervised
Unsupervised
Regression (continuous outcome)
Classification (categorical outcome)
Survival outcome
Clustering (grouping observations)
Data Reduction (transforming variables)