# Outline for Part 2

Measuring prediction performance

Sample splitting

Resampling methods

# Which model is best for prediction?

## Example: Regularization/Variable selection by Lasso

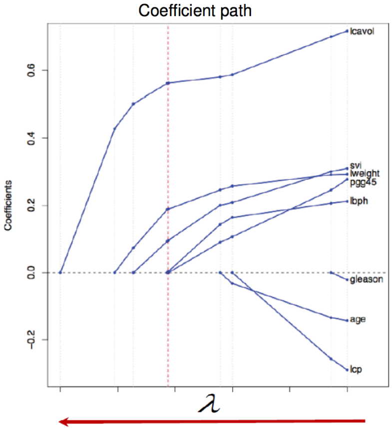
**Idea:**
Penalize (shrink towards zero) regression coefficients by adding penalty term to LS criterion.

Thereby, "non-relevant" coefficients are estimated as exactly 0 and can be excluded.

$$\hat{\beta}^{\text{lasso}} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left( y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

Penalty controlled by regularization parameter $\lambda$:

- small $\lambda \Rightarrow$ many variables in model
- large $\lambda \Rightarrow$ few variables in model




$\Rightarrow$ How to select $\lambda$ to minimize prediction error?

# Measuring prediction performance

To evaluate model performance on a given data set, measure how well its predictions actually match the observed data.

How close is the predicted value to the true value for that observation?

- **Linear Regression:** Mean squared error:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- **2-class Classification:** Brier score:

$$BS = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{p}(y_i = 1|x_i))^2$$

# Performance measures

Some models are used only for parameter estimation and testing

But:

- If used for prediction/classification, need to consider accuracy of predictions
- Two major aspects of prediction accuracy that need to be assessed:

  (1) Reliability or calibration of a model:
    - ability of the model to make unbiased estimates of the outcome
    - observed responses agree with predicted responses

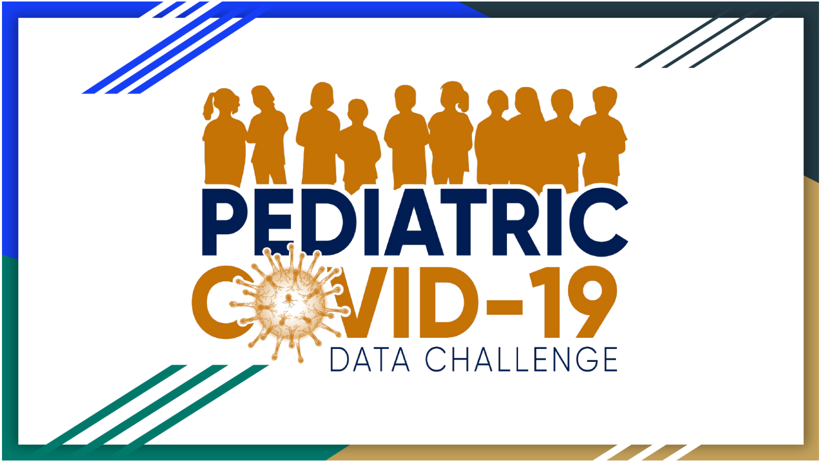
  (2) Discrimination ability:
    - the model is able, through the use of predicted responses, to separate subjects

# Performance measures for classification tasks

## Steyerberg et al, 2010 (Table 1)

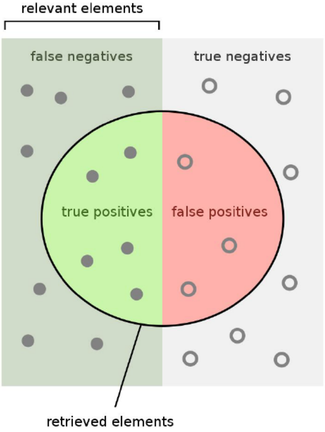
| Aspect | Measure | Visualization | Characteristics |
|---|---|---|---|
| Overall performance | $R^2$ Brier → Brier score | Validation graph | Better with lower distance between Y and $\hat{Y}$. Captures calibration and discrimination aspects. |
| Discrimination | C statistic → AUC | ROC curve | Rank order statistic; Interpretation for a pair of patients with and without the outcome |
| | Discrimination slope | Box plot | Difference in mean of predictions between outcomes; Easy visualization |
| Calibration | Calibration-in-the-large | Calibration or validation graph | Compare mean(y) versus mean($\hat{y}$); essential aspect for external validation |
| | Calibration slope | | Regression slope of linear predictor; essential aspect for internal and external validation related to 'shrinkage' of regression coefficients |
| | Hosmer-Lemeshow test | | Compares observed to predicted by decile of predicted probability |

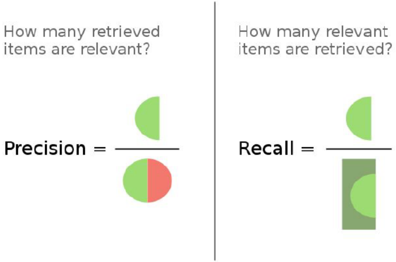
# Example: Data challenge model performance evaluation



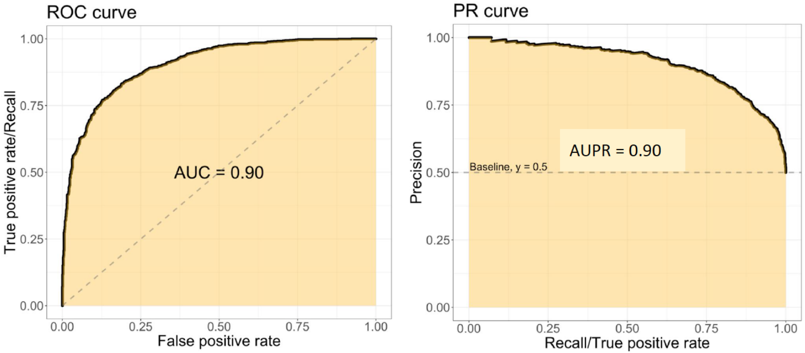

https://drive.hhs.gov/pediatric_challenge.html

# Example: Data challenge model performance evaluation



# Example: Data challenge model performance evaluation




$$F_\beta = (1 + \beta^2) \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{(\beta^2 \cdot \mathrm{precision}) + \mathrm{recall}}$$

# Example: Data challenge model performance evaluation

Quantitative score (85 %):

$$\frac{1}{3}\left(\left(\max_{\text{threshold } t} F_2(t)\right)^2 + \mathrm{AUPR}^2 + \left(\mathrm{Mean}(\mathrm{AUROC}) - \mathrm{Var}(\mathrm{AUROC})\right)^2\right)$$

Qualitative score (15 %):

- Timeliness
- Interpretability
- Context Utility
- Technical Reproducibility
- Prediction Reproducibility

How to estimate the performance measure in an unbiased manner?

# How to estimate performance in an unbiased manner?

**Need:** Model assessment/validation to ascertain whether predicted values from the model are likely to accurately predict responses on future subjects or subjects not used to develop the model

**Two modes of validation**

- External:
  Use different sets of subjects for building the model (including tuning) and testing
- Internal:
  (i) Apparent (or training) error: evaluate fit on same data used to create fit
  (ii) Data splitting and its extensions
  (iii) Resampling methods

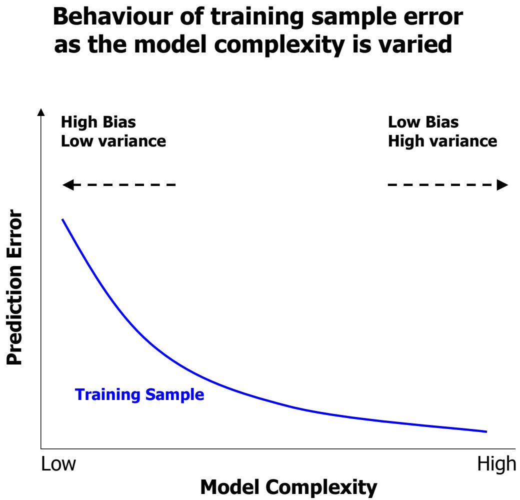
- **Two fundamental problems with estimation on the training data:**
  - The final model will over-fit the training data. Problem is more pronounced with models with a large number of variables.
  - The error estimate will be overly optimistic (too low).

- A much better idea is to **split the data** into disjoint subsets or use **resampling methods**

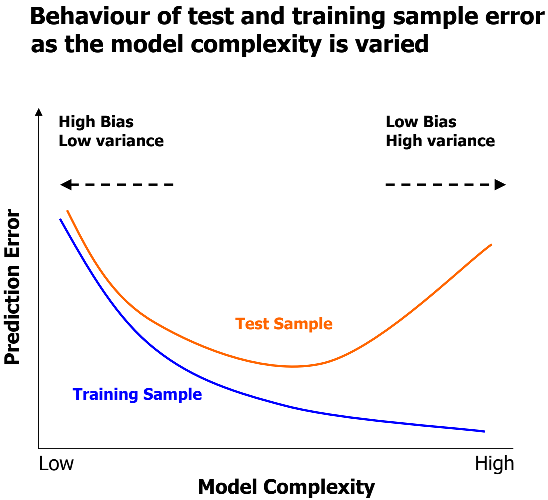
- **Training error**: Classification error in the training data set
- **Generalisation error**: Expected error for the classification of new samples $\rightarrow$ This is what we want to estimate!

The training error is a bad estimator for the generalisation error!

# Over-fitting is a major problem



# Over-fitting is a major problem



# The Bias-Variance Trade-Off

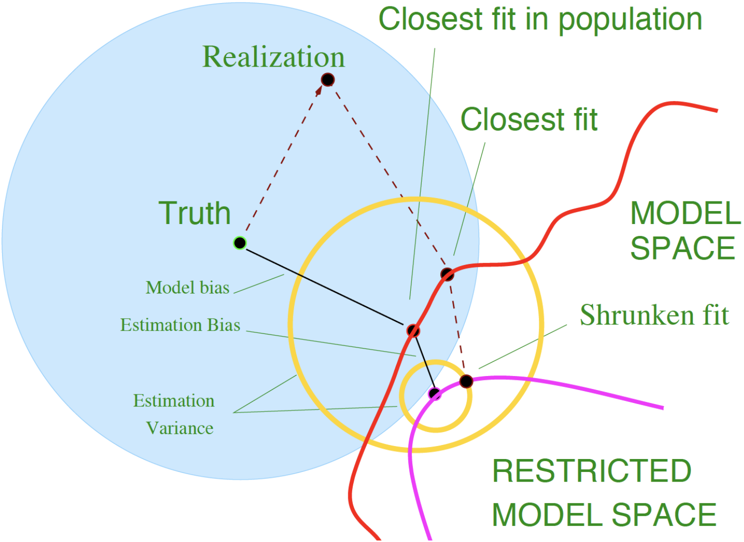
- A simple model might have more model bias, but
- A complex model has more model variance.

For $Y = f(X) + \epsilon$ with $E(\epsilon) = 0$ and $Var(\epsilon) = \sigma_\epsilon^2$, the expected prediction error of $\hat{f}(X)$ at point $x_0$ with squared error loss is:

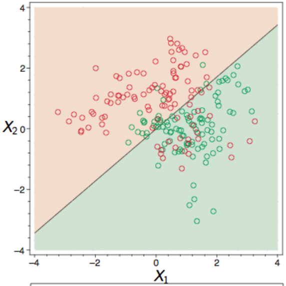
$$\begin{aligned} \mathrm{Err}(x_0) &= E[(Y - \hat{f}(x_0))^2 | X = x_0] \\ &= \sigma_\varepsilon^2 + [\mathrm{E}\hat{f}(x_0) - f(x_0)]^2 + E[\hat{f}(x_0) - \mathrm{E}\hat{f}(x_0)]^2 \\ &= \sigma_\varepsilon^2 + \mathrm{Bias}^2(\hat{f}(x_0)) + \mathrm{Var}(\hat{f}(x_0)) \\ &= \text{Irreducible Error} + \mathrm{Bias}^2 + \text{Variance}. \qquad (7.9) \end{aligned}$$

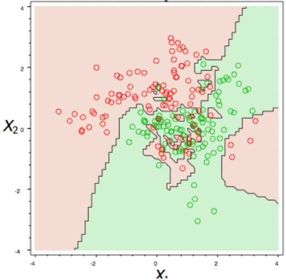
from Hastie et al. (2009), chapter 7.3

# The Bias-Variance Trade-Off



# The danger for over-fitting is higher with complex models



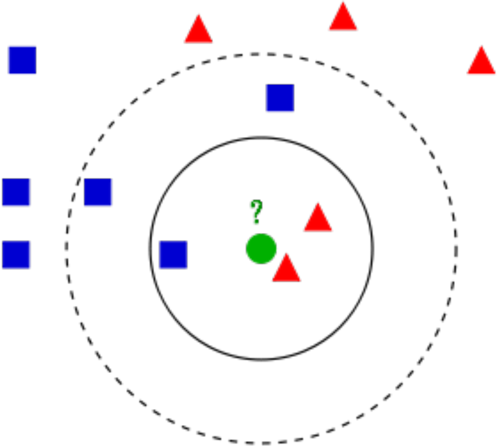

**Linear model**

- Low complexity
- Stable (linear) decision boundary
- Generalisation error might be hardly larger than the training error

**1-Nearest-neighbour method**

- High complexity
- Unstable (highly non-linear) decision boundary
- Large over-fitting likely: Generalisation error probably much larger than training error

# k-Nearest-neighbour method




Wikipedia.org

- k=3: Classify the test sample as a red triangle.
- k=5: Classify the test sample as a blue square.

# Model building, selection and assessment

1. How to decide which method is the "best", i.e. has the smallest generalisation error, in a specific situation?
2. And how large is that smallest generalisation error anyway?

- **Model building and selection**: For a variety of different methods
  1. Fit ("train") the models,
     i.e. perform parameter tuning/ variable selection
  2. Estimate the prediction errors.
  3. Choose the "best" method for a specific situation.

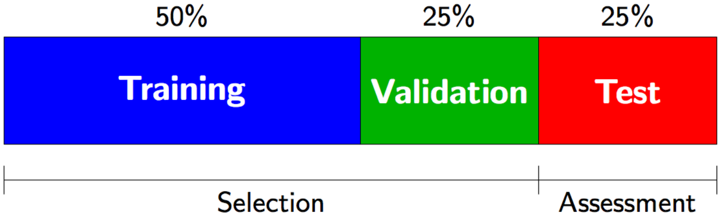
- **Model assessment**
  - For the final selected model estimate the generalisation error on *new data*.

# Sample splitting

→ Split data in several independent subsets before model building.

# Sample splitting

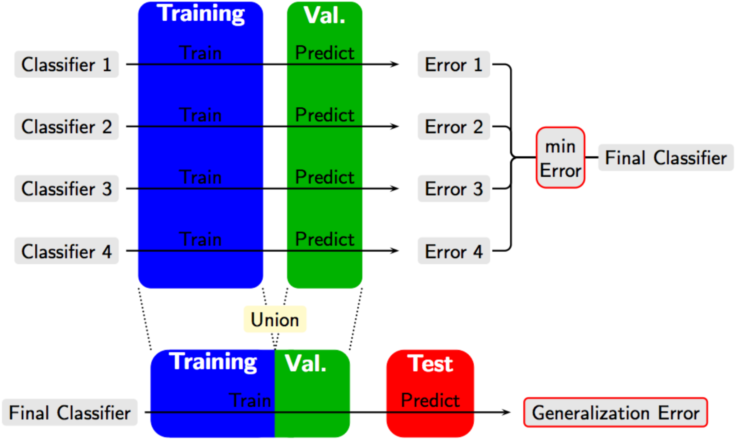
In a data-rich situation, we can split the available data.




- **Training set**: Fit ("train") the various prediction models
- **Validation set**:
  - Estimate the prediction errors of the models
  - Final model: Choose model with smallest prediction error
- **Test set**: Estimate the generalisation error by applying the final model to a new test data set

# Sample splitting

## Model building and selection →




## → Model assessment

# Drawbacks of sample splitting

One-time sample splitting has two **basic drawbacks**:

- We may not be able to afford the "luxury" of setting aside a portion of the data set for testing, as it might result in a large loss of power.
- The assessment can vary greatly when taking different splits:

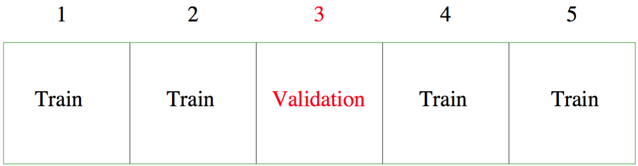
  Since it is a single train-and-test experiment, the estimate of the error rate will be misleading if we happen to get an "unfortunate" split.

# Resampling methods

- → Cross-validation
- → Bootstrapping

# Cross-validation

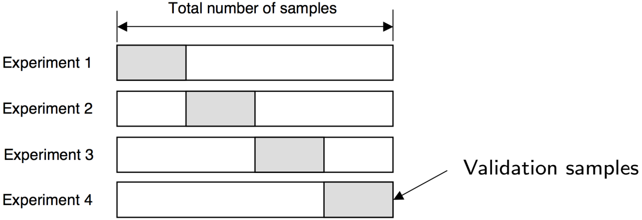
- Alternative to data splitting in not so data-rich situations (i.e. most of the time...)
- Partition the data set into $K$ roughly equal-sized subsets
- Each subset will be the test data set once, with the remaining samples making up the training data



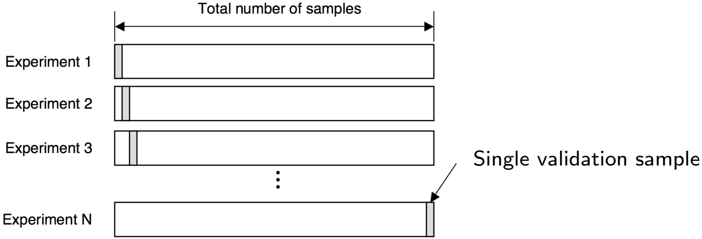

- Cross-validation error: The results are pooled from all test sets to estimate the performance of the model (each case is used exactly once).

# Cross-validation

- *K*-fold cross-validation

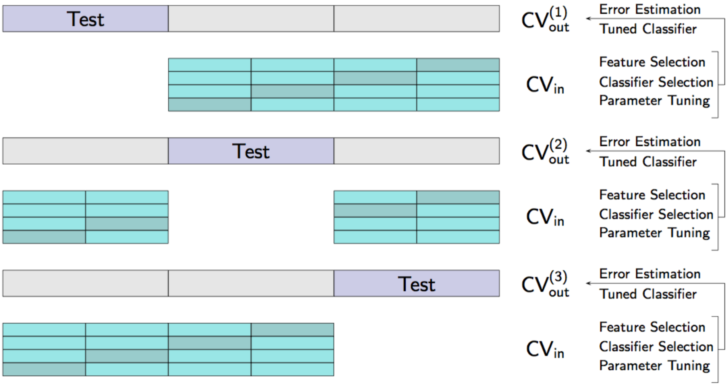


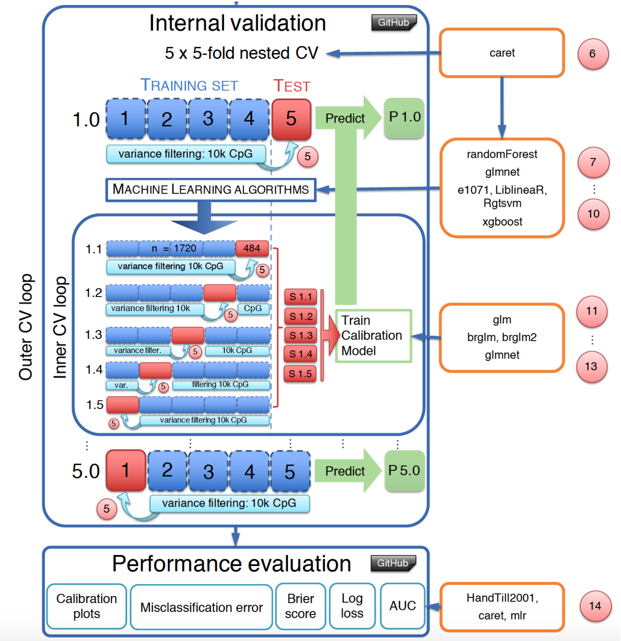

- Leave-one-out cross-validation



# Nested cross-validation

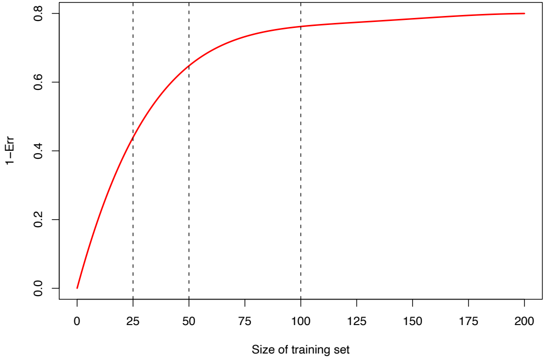
- Inner CV loop: Model building and selection
  - Feature selection, model selection, parameter tuning
  - Choose the model with the smallest CV error within inner loop
- Outer CV loop: Model assessment
  - Estimate the generalisation error for the final model






from: Maros et al. (2020)

# K-fold cross-validation: Training set size bias




**Hypothetical learning curve:**
The performance of the predictor improves as the training set size increases to about 100 observations.

Increasing this number further brings only a small benefit.

# Drawbacks of cross-validation

- **Leave-one-out CV:** may have large variance
- **K-fold CV:** may have large bias, depending on the choice of the number of observations to be held out from each fit. The bias is possibly severe for training set sizes $< 50$, say. If the learning curve has a considerable slope at the given training set size, 5 or 10-fold CV will strongly overestimate the true prediction error.
- Possible solution: estimate prediction error by **bootstrapping**