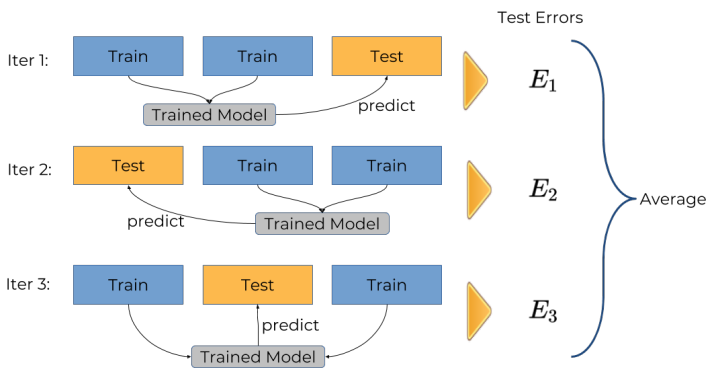
Introduction to Machine Learning

Evaluation: Resampling 1

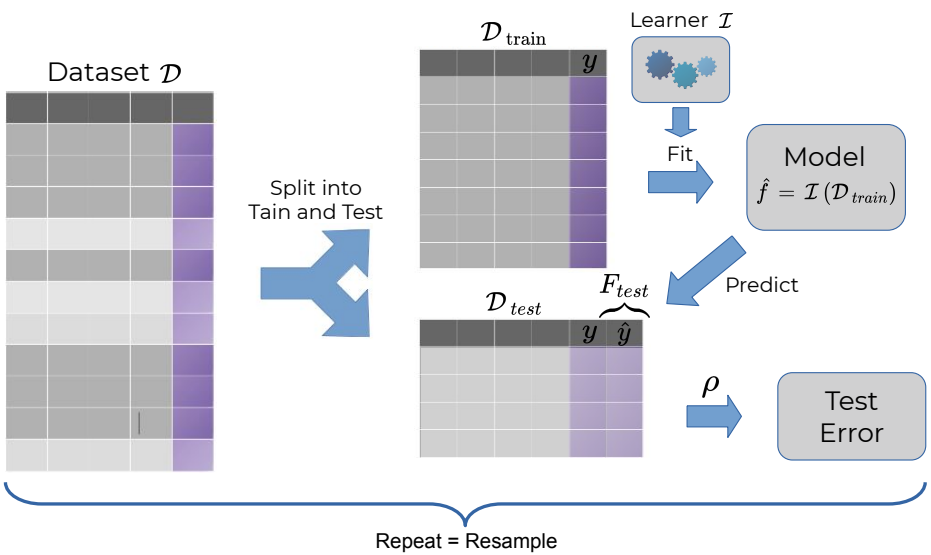


Learning goals

- Understand how resampling techniques extend the idea of simple train-test splits
- Understand the ideas of cross-validation, bootstrap and subsampling
- Understand what pessimistic bias means

RESAMPLING

- **Goal**: estimate $\mathrm{GE}(\mathcal{I}, \boldsymbol{\lambda}, n, \rho) = \lim_{n_{\mathrm{test}} \to \infty} \mathbb{E}\left[\rho\left(\mathbf{y}, \boldsymbol{F}_{\mathcal{D}_{\mathrm{test}}, \mathcal{I}(\mathcal{D}_n, \boldsymbol{\lambda})}\right)\right]$.
- Use the data at hand efficiently.
- Repeatedly split in train and test, then average results.
- Make training sets large (to keep pessimistic bias small since we use $\mathrm{GE}(\mathcal{I}, \boldsymbol{\lambda}, n_{\mathrm{train}}, \rho)$ as a proxy for $\mathrm{GE}(\mathcal{I}, \boldsymbol{\lambda}, n, \rho)$), and reduce variance introduced by smaller test sets through many repetitions / averaging of results.
- Problems with single-holdout split: small train set leads to high pessimistic bias and small test set results in high variance.



RESAMPLING STRATEGIES

- Different **resampling strategies** exist to balance bias against variance, e.g., holdout sampling or cross-validation.
- To ease notation, we represent our train and test sets by index vectors $J_{\text{train}} \in \{1, \ldots, n\}^{n_{\text{train}}}$ and $J_{\text{test}} \in \{1, \ldots, n\}^{n_{\text{test}}}$, and define a resampling strategy with B train-test splits by

$$\mathcal{J} = \left((J_{\text{train},1}, J_{\text{test},1}), \ldots, (J_{\text{train},B}, J_{\text{test},B}) \right).$$

- Based on $\mathcal{J}$, we can express our estimate of the **generalization error** $\text{GE}(\mathcal{I}, \boldsymbol{\lambda}, n_{\text{train}}, \rho)$ for arbitrary resampling strategies as

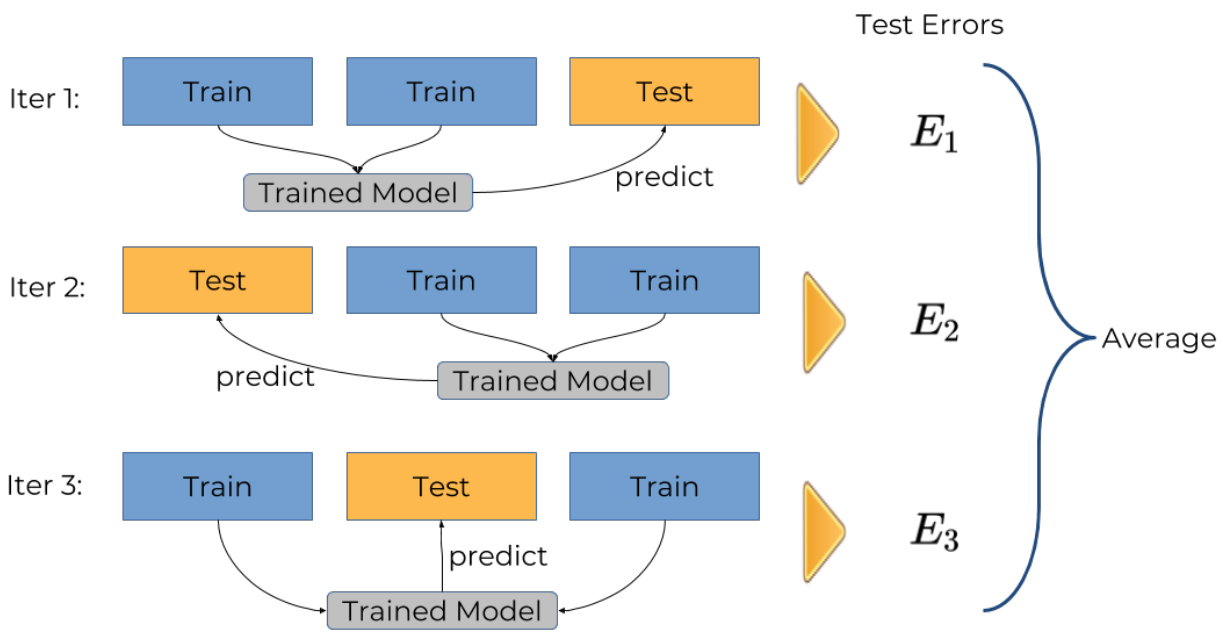
$$\widehat{\text{GE}}(\mathcal{I}, \mathcal{J}, \rho, \boldsymbol{\lambda}) = \text{agr}\Bigg(\rho\Big(\mathbf{y}_{J_{\text{test},1}}, \boldsymbol{F}_{J_{\text{test},1}, \mathcal{I}(\mathcal{D}_{\text{train},1}, \boldsymbol{\lambda})} \Big), \\ \vdots \\ \rho\Big(\mathbf{y}_{J_{\text{test},1}}, \boldsymbol{F}_{J_{\text{test},B}, \mathcal{I}(\mathcal{D}_{\text{train},B}, \boldsymbol{\lambda})} \Big) \Bigg),$$

where the aggregation agr is typically chosen to be the mean and $n_{\text{train}} \approx n_{\text{train},1} \approx \cdots \approx n_{\text{train},B}$.

CROSS-VALIDATION

- Split the data into k roughly equally-sized partitions.
- Use each part once as test set and join the respective $k - 1$ others for training.
- Obtain k test errors and average.

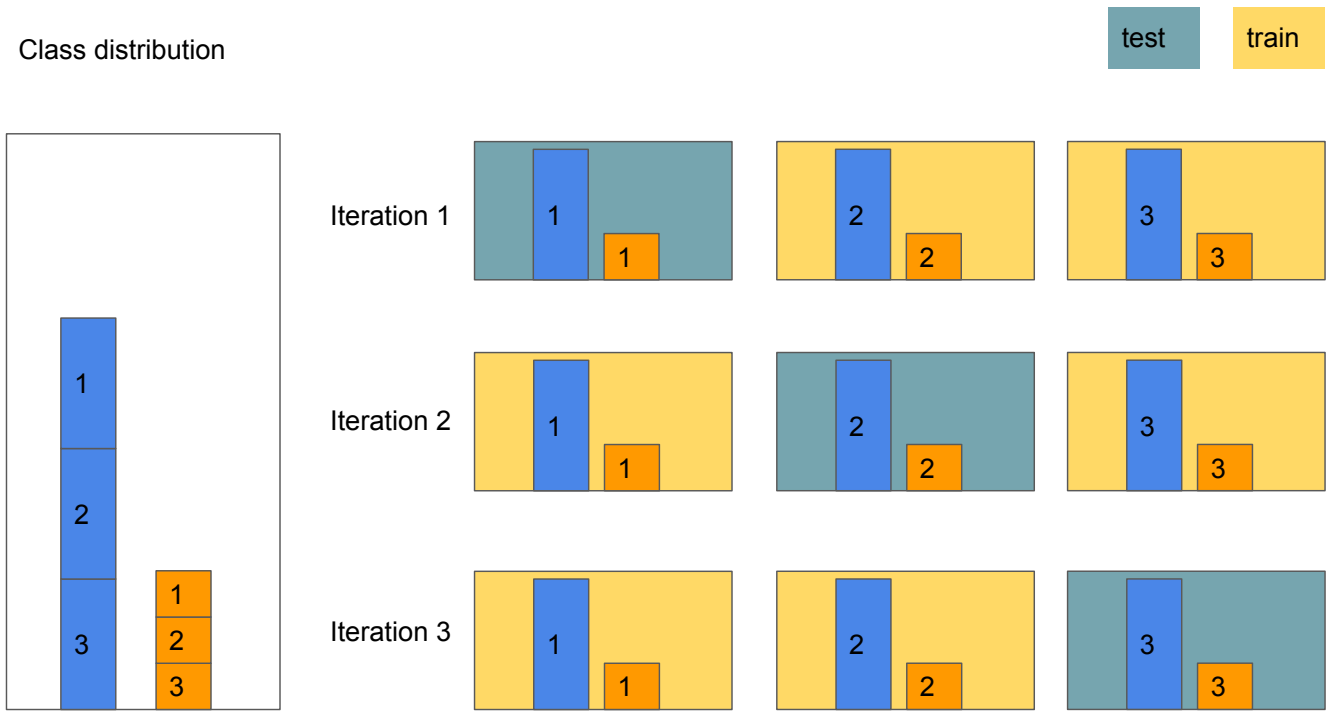
Example: 3-fold cross-validation



CROSS-VALIDATION - STRATIFICATION

Stratification attempts to preserve the distribution of the target class (or any specific categorical feature of interest) in each fold.

Example: stratified 3-fold cross-validation

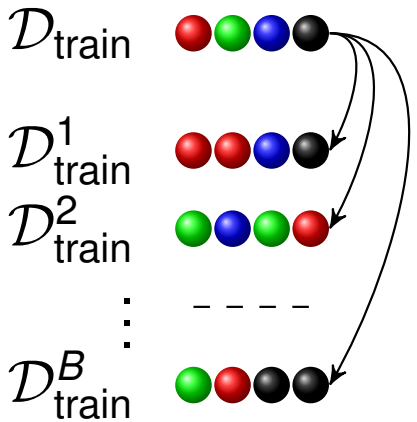


CROSS-VALIDATION

- 5 or 10 folds are common.
- $k = n$ is known as leave-one-out (LOO) cross-validation.
- GE estimates tend to be pessimistically biased: size of the training sets is $n - \frac{n}{k} < n$.
  $\Rightarrow$ Bias increases as k gets smaller.
- The k performance estimates are dependent because of the structured overlap of the training sets.
  $\Rightarrow$ Variance of the estimator increases for very large k (approaching LOO), when training sets nearly completely overlap.
- Repeated k-fold CV (multiple random partitions) can improve error estimation for small sample sizes.

BOOTSTRAP

The basic idea is to randomly draw B training sets of size n with replacement from the original training set $\mathcal{D}_{\text{train}}$:



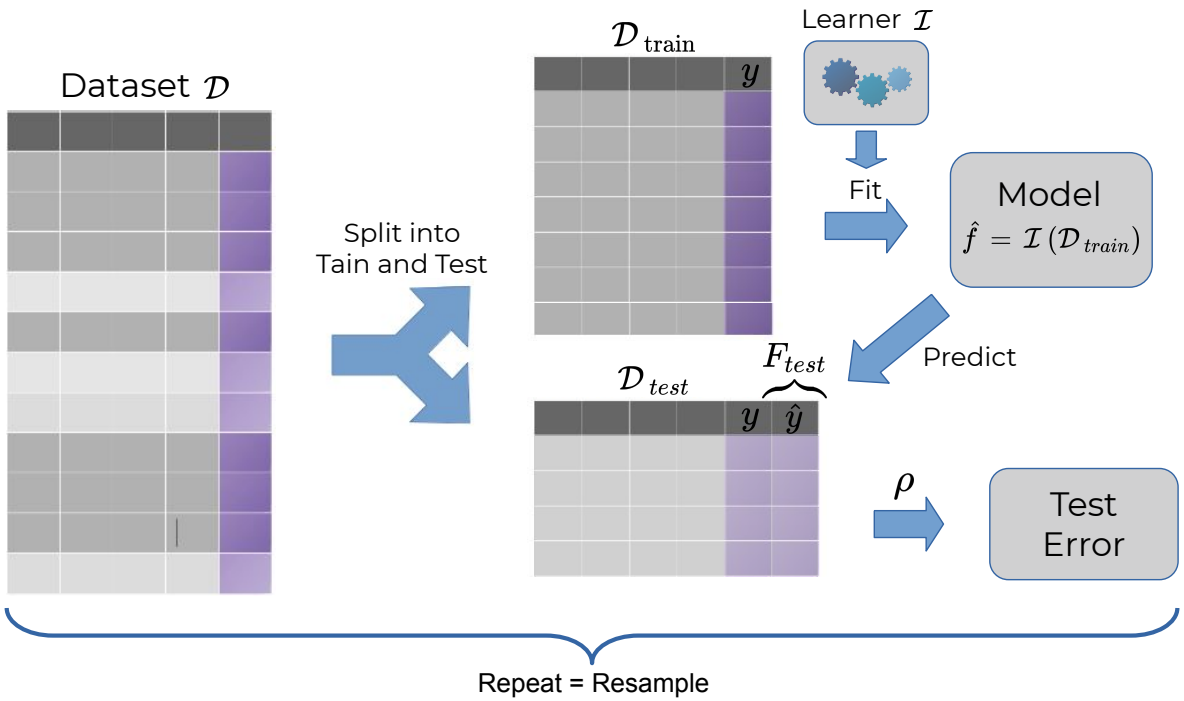
We define the test set in terms of out-of-bag observations $\mathcal{D}^b_{\text{test}} = \mathcal{D}_{\text{train}} \setminus \mathcal{D}^b_{\text{train}}$, $b = 1, 2, \ldots, B$.

BOOTSTRAP

- Typically, B is between 30 and 200.
- The variance of the bootstrap estimator tends to be smaller than the variance of k-fold CV.
  $\Rightarrow$ More iterations, smaller variance.
- As in k-fold CV, GE estimates tend to be pessimistically biased (because training sets contain only approximately about $1 - \mathbb{P}((\mathbf{x}, y) \notin \mathcal{D}_{\text{train}}) = 1 - \left(1 - \frac{1}{n}\right)^n \overset{n \to \infty}{\longrightarrow} 1 - \frac{1}{e} \approx 63.2\%$ of the unique observations).
- Bootstrapping framework allows for inference (e.g., detecting significant performance differences between learners).
- Extensions exist for very small data sets that also use the training error for estimation: B632 and B632+.

SUBSAMPLING

- Repeated hold-out with averaging, a.k.a. Monte Carlo CV.
- Similar to bootstrap, but draws without replacement.
- Typical choices for splitting: $\frac{4}{5}$ or $\frac{9}{10}$ for training.



- The smaller the subsampling rate, the larger the pessimistic bias.
- The more subsampling repetitions, the smaller the variance.

LEAVE-ONE-OBJECT-OUT

In the situation where our data consist of multiple observations originating from the same instances we need to adapt our resampling strategy. (Note: In this case the observations are not i.i.d)

One simple solution is to train on all observations originating from all instances except one and then evaluate on the observations of the remaining one. This procedure is called leave-one-object-out.