# T3: Model evaluation

Fundamentos del Aprendizaje Automático

Curso 2025/2026

# Structure

# Outline

# Principles

- So far: **Infer knowledge** (*train* model) and **predict** (*test* model)

# Principles

- So far: **Infer knowledge** (*train* model) and **predict** (*test* model)
  - $\rightarrow$ How well is it performing? $\Rightarrow$ Model evaluation

# Principles

- So far: **Infer knowledge** (*train* model) and **predict** (*test* model)
  - $\rightarrow$ How well is it performing? $\Rightarrow$ Model evaluation

- Is model evaluation connected to loss, risk, and error?

# Principles

- So far: **Infer knowledge** (*train* model) and **predict** (*test* model)
  → How well is it performing? ⇒ Model evaluation

- Is model evaluation connected to loss, risk, and error?
  → Related but not the same:

| **Concept** | **Phase** | **Goal** | **Module** | **Meaning** |
|---|---|---|---|---|
| Loss | Train | Guide the optimization process | T2 (Computational learning) | Error on a single sample |
| Risk[1] | | | | Expected loss value across the data distribution |
| Evaluation | Test | Quantify the performance of the model | T3 (Model evaluation) | How well the model performs on unseen data |

[1] Equals *error* considering a zero-one loss function.

# True, expected, and empirical risk/error

# True, expected, and empirical risk/error

- True risk/error: the expected loss under true distribution $P(\mathbf{x}, \omega)$:

# True, expected, and empirical risk/error

- True risk/error: the expected loss under true distribution $P(\mathbf{x}, \omega)$:

$$R(\gamma) = \mathbb{E}_{(\mathbf{x},\omega)\sim P(\mathbf{x},\omega)}\left[\lambda(\gamma(\mathbf{x})|\omega)\right]$$

$\Rightarrow$ Purely theoretical (not computable)

# True, expected, and empirical risk/error

- True risk/error: the expected loss under true distribution $P(\mathbf{x}, \omega)$:

$$R(\gamma) = \mathbb{E}_{(\mathbf{x},\omega)\sim P(\mathbf{x},\omega)} \left[\lambda(\gamma(\mathbf{x})|\omega)\right]$$

  $\Rightarrow$ Purely theoretical (not computable)

- Expected risk/error: expected *true error* averaged over all possible training sets $\mathcal{D}$ that could be drawn from $P(\mathbf{x}, \omega)$:

# True, expected, and empirical risk/error

- True risk/error: the expected loss under true distribution $P(\mathbf{x}, \omega)$:

$$R(\gamma) = \mathbb{E}_{(\mathbf{x},\omega)\sim P(\mathbf{x},\omega)} \left[\lambda(\gamma(\mathbf{x})|\omega)\right]$$

$\Rightarrow$ Purely theoretical (not computable)

- Expected risk/error: expected *true error* averaged over all possible training sets $\mathcal{D}$ that could be drawn from $P(\mathbf{x}, \omega)$:

$$R_{\mathcal{D}}(\gamma) = \mathbb{E}_{(\mathbf{x}_i,\omega_i)\in\mathcal{D}\sim P(\mathbf{x},\omega)} \left[\lambda(\gamma(\mathbf{x}_i; \mathcal{D})|\omega_i)\right]$$

$\Rightarrow$ Purely theoretical (not computable)

# True, expected, and empirical risk/error

- True risk/error: the expected loss under true distribution $P(\mathbf{x}, \omega)$:

$$R(\gamma) = \mathbb{E}_{(\mathbf{x},\omega)\sim P(\mathbf{x},\omega)}\left[\lambda(\gamma(\mathbf{x})|\omega)\right]$$

$\Rightarrow$ Purely theoretical (not computable)

- Expected risk/error: expected *true error* averaged over all possible training sets $\mathcal{D}$ that could be drawn from $P(\mathbf{x}, \omega)$:

$$R_{\mathcal{D}}(\gamma) = \mathbb{E}_{(\mathbf{x}_i,\omega_i)\in\mathcal{D}\sim P(\mathbf{x},\omega)}\left[\lambda(\gamma(\mathbf{x}_i;\mathcal{D})|\omega_i)\right]$$

$\Rightarrow$ Purely theoretical (not computable)

- Empirical risk/error: estimation of the *true/expected error* using a finite data collection $\mathcal{D} = \left\{(\mathbf{x}_1, \omega_1), \ldots, (\mathbf{x}_{|\mathcal{D}|}, \omega_{|\mathcal{D}|})\right\}$:

# True, expected, and empirical risk/error

- True risk/error: the expected loss under true distribution $P(\mathbf{x},\omega)$:

$$R(\gamma) = \mathbb{E}_{(\mathbf{x},\omega)\sim P(\mathbf{x},\omega)}\left[\lambda(\gamma(\mathbf{x})|\omega)\right]$$

$\Rightarrow$ Purely theoretical (not computable)

- Expected risk/error: expected *true error* averaged over all possible training sets $\mathcal{D}$ that could be drawn from $P(\mathbf{x},\omega)$:

$$R_{\mathcal{D}}(\gamma) = \mathbb{E}_{(\mathbf{x}_i,\omega_i)\in\mathcal{D}\sim P(\mathbf{x},\omega)}\left[\lambda(\gamma(\mathbf{x}_i;\mathcal{D})|\omega_i)\right]$$

$\Rightarrow$ Purely theoretical (not computable)

- Empirical risk/error: estimation of the *true/expected error* using a finite data collection $\mathcal{D} = \left\{(\mathbf{x}_1,\omega_1),\ldots,(\mathbf{x}_{|\mathcal{D}|},\omega_{|\mathcal{D}|})\right\}$:

$$\hat{R}(\gamma) = \frac{1}{|\mathcal{D}|}\sum_{(\mathbf{x}_i,\omega_i)\in\mathcal{D}} \lambda\left(\gamma(\mathbf{x}_i)|\omega_i\right)$$

$\Rightarrow$ Practical and computable

# (Training) Loss and (evaluation) metric/figure of merit

There exist two levels of relationship:

# (Training) Loss and (evaluation) metric/figure of merit

There exist two levels of relationship:

1. **Aligned** case: Training loss and evaluation metric match
2. **Misaligned** case: Training loss and evaluation metric differ

# (Training) Loss and (evaluation) metric/figure of merit

There exist two levels of relationship:

1. **Aligned** case: Training loss and evaluation metric match
2. **Misaligned** case: Training loss and evaluation metric differ
   - **Loss** is a *proxy* easier to optimize
   - **Metric** measures *real-world* performance

# Example: Weather prediction

A client reaches two AI engineers to design a wheather prediction system that works on a daily basis:

# Example: Weather prediction

A client reaches two AI engineers to design a wheather prediction system that works on a daily basis:

1. Feature extraction $\Rightarrow \mathbf{x} \in \mathbb{R}^d$

# Example: Weather prediction

A client reaches two AI engineers to design a wheather prediction system that works on a daily basis:

1. Feature extraction $\Rightarrow \mathbf{x} \in \mathbb{R}^d$
2. Two states of nature $\Rightarrow \mathcal{W} = \{\text{sunny}, \text{rainy}\}$

# Example: Weather prediction

A client reaches two AI engineers to design a wheather prediction system that works on a daily basis:

1. Feature extraction $\Rightarrow \mathbf{x} \in \mathbb{R}^d$
2. Two states of nature $\Rightarrow \mathcal{W} = \{\text{sunny}, \text{rainy}\}$
3. Labeled dataset $\Rightarrow \mathcal{D} = \mathcal{D}_s \cup \mathcal{D}_r$ with $|\mathcal{D}| = 365$ days

# Example: Weather prediction

A client reaches two AI engineers to design a wheather prediction system that works on a daily basis:

1. Feature extraction $\Rightarrow \mathbf{x} \in \mathbb{R}^d$
2. Two states of nature $\Rightarrow \mathcal{W} = \{\text{sunny}, \text{rainy}\}$
3. Labeled dataset $\Rightarrow \mathcal{D} = \mathcal{D}_s \cup \mathcal{D}_r$ with $|\mathcal{D}| = 365$ days
   - $\mathcal{D}_s = \{(\mathbf{x}_1, \text{s}), \ldots, (\mathbf{x}_{355}, \text{s})\}$
   - $\mathcal{D}_r = \{(\mathbf{x}_{356}, \text{r}), \ldots, (\mathbf{x}_{365}, \text{r})\}$

# Example: Weather prediction

A client reaches two AI engineers to design a wheather prediction system that works on a daily basis:

1. Feature extraction $\Rightarrow \mathbf{x} \in \mathbb{R}^d$
2. Two states of nature $\Rightarrow \mathcal{W} = \{\text{sunny}, \text{rainy}\}$
3. Labeled dataset $\Rightarrow \mathcal{D} = \mathcal{D}_s \cup \mathcal{D}_r$ with $|\mathcal{D}| = 365$ days
   - $\mathcal{D}_s = \{(\mathbf{x}_1, \text{s}), \ldots, (\mathbf{x}_{355}, \text{s})\}$
   - $\mathcal{D}_r = \{(\mathbf{x}_{356}, \text{r}), \ldots, (\mathbf{x}_{365}, \text{r})\}$
4. Procedure:
   - Use $\mathcal{D}$ to adjust and evaluate
   - **Metric**: number of correct predictions

# Example: Weather prediction

| **Engineer #1** | **Engineer #2** |
| --- | --- |

# Example: Weather prediction

**Engineer #1**

**Policy**: Adequate study/design

**Engineer #2**

**Policy**: Always predicts sunny

# Example: Weather prediction

**Engineer #1**

**Policy**: Adequate study/design

- Estimations for $\mathcal{D}_s$:
  - ✓ 350 as sunny
  - ✗ 5 as rainy
- Estimations for $\mathcal{D}_r$:
  - ✗ 5 as sunny
  - ✓ 5 as rainy

**Engineer #2**

**Policy**: Always predicts sunny

# Example: Weather prediction

**Engineer #1**

**Policy**: Adequate study/design

- Estimations for $\mathcal{D}_s$:
  - ✓ 350 as sunny
  - ✗ 5 as rainy
- Estimations for $\mathcal{D}_r$:
  - ✗ 5 as sunny
  - ✓ 5 as rainy

$$\text{Acc}_{\#1} = \frac{350 + 5}{365} \approx 97\%$$

**Engineer #2**

**Policy**: Always predicts sunny

# Example: Weather prediction

**Engineer #1**

**Policy**: Adequate study/design

- Estimations for $\mathcal{D}_s$:
  - ✓ 350 as sunny
  - ✗ 5 as rainy
- Estimations for $\mathcal{D}_r$:
  - ✗ 5 as sunny
  - ✓ 5 as rainy

$$\mathrm{Acc}_{\#1} = \frac{350 + 5}{365} \approx 97\%$$

**Engineer #2**

**Policy**: Always predicts sunny

- Estimations for $\mathcal{D}_s$:
  - ✓ 355 as sunny
  - ✓ 0 as rainy
- Estimations for $\mathcal{D}_r$:
  - ✗ 10 as sunny
  - ✗ 0 as rainy

# Example: Weather prediction

**Engineer #1**

**Policy**: Adequate study/design

- Estimations for $\mathcal{D}_s$:
  - ✓ 350 as sunny
  - ✗ 5 as rainy
- Estimations for $\mathcal{D}_r$:
  - ✗ 5 as sunny
  - ✓ 5 as rainy

$$\mathrm{Acc}_{\#1} = \frac{350 + 5}{365} \approx 97\%$$

**Engineer #2**

**Policy**: Always predicts sunny

- Estimations for $\mathcal{D}_s$:
  - ✓ 355 as sunny
  - ✓ 0 as rainy
- Estimations for $\mathcal{D}_r$:
  - ✗ 10 as sunny
  - ✗ 0 as rainy

$$\mathrm{Acc}_{\#2} = \frac{355 + 0}{365} \approx 97\%$$

# Example: Weather prediction

**Engineer #1**

**Policy**: Adequate study/design

- Estimations for $\mathcal{D}_s$:
  - ✓ 350 as sunny
  - ✗ 5 as rainy
- Estimations for $\mathcal{D}_r$:
  - ✗ 5 as sunny
  - ✓ 5 as rainy

$$\text{Acc}_{\#1} = \frac{350 + 5}{365} \approx 97\%$$

**Engineer #2**

**Policy**: Always predicts sunny

- Estimations for $\mathcal{D}_s$:
  - ✓ 355 as sunny
  - ✓ 0 as rainy
- Estimations for $\mathcal{D}_r$:
  - ✗ 10 as sunny
  - ✗ 0 as rainy

$$\text{Acc}_{\#2} = \frac{355 + 0}{365} \approx 97\%$$

Which is the **issue** here?

# Outline

# Introduction

# Introduction

- In practical scenarios $\Rightarrow$ finite set of data $\mathcal{D} = \{(\mathbf{x}_i, \omega_i)\}_{i=1}^{|\mathcal{D}|}$

# Introduction

- In practical scenarios $\Rightarrow$ finite set of data $\mathcal{D} = \{(\mathbf{x}_i, \omega_i)\}_{i=1}^{|\mathcal{D}|}$
- Assortment $\mathcal{D}$ is used for several procedures in the model creation:

# Introduction

- In practical scenarios $\Rightarrow$ finite set of data $\mathcal{D} = \{(\mathbf{x}_i, \omega_i)\}_{i=1}^{|\mathcal{D}|}$

- Assortment $\mathcal{D}$ is used for several procedures in the model creation:
  1. **Train**: Provide data for the model to learn
  2. **Tune**: Obtain the most adequate set of (hyper)parameters
  3. **Assess**: Evaluate the goodness of the model

# Introduction

- In practical scenarios $\Rightarrow$ finite set of data $\mathcal{D} = \{(\mathbf{x}_i, \omega_i)\}_{i=1}^{|\mathcal{D}|}$

- Assortment $\mathcal{D}$ is used for several procedures in the model creation:
  1. **Train**: Provide data for the model to learn
  2. **Tune**: Obtain the most adequate set of (hyper)parameters
  3. **Assess**: Evaluate the goodness of the model

  $\Rightarrow$ *Entire dataset? Subsets?*

# Introduction

- In practical scenarios $\Rightarrow$ finite set of data $\mathcal{D} = \{(\mathbf{x}_i, \omega_i)\}_{i=1}^{|\mathcal{D}|}$

- Assortment $\mathcal{D}$ is used for several procedures in the model creation:
  1. **Train**: Provide data for the model to learn
  2. **Tune**: Obtain the most adequate set of (hyper)parameters
  3. **Assess**: Evaluate the goodness of the model

  $\Rightarrow$ *Entire dataset? Subsets?*

- Typical strategies:

# Introduction

- In practical scenarios $\Rightarrow$ finite set of data $\mathcal{D} = \{(\mathbf{x}_i, \omega_i)\}_{i=1}^{|\mathcal{D}|}$
- Assortment $\mathcal{D}$ is used for several procedures in the model creation:
  1. **Train**: Provide data for the model to learn
  2. **Tune**: Obtain the most adequate set of (hyper)parameters
  3. **Assess**: Evaluate the goodness of the model

  $\Rightarrow$ *Entire dataset? Subsets?*
- Typical strategies:
  1. **Data partitioning**: Adequately divide assortment $\mathcal{D}$
  2. **Cross-validation procedures**: Exhaustively explore assortment $\mathcal{D}$

# Formalization

# Formalization

- **Premise**: The procedures must use different data partitions

# Formalization

- **Premise**: The procedures must use different data partitions
  - $\rightarrow$ Otherwise, data leakage $\Rightarrow$ Optimistic estimates

# Formalization

- **Premise**: The procedures must use different data partitions
  - $\rightarrow$ Otherwise, data leakage $\Rightarrow$ Optimistic estimates

- **Subsets** of the assortment: $\mathcal{D} = \underbrace{\mathcal{D}_{train}}_{Train} \cup \underbrace{\mathcal{D}_{validation}}_{Tune} \cup \underbrace{\mathcal{D}_{test}}_{Assess}$
  - $\rightarrow$ Disjoint subsets: $\mathcal{D}_{train} \cap \mathcal{D}_{validation} \cap \mathcal{D}_{test} = \emptyset$

# Formalization

- **Premise**: The procedures must use different data partitions
  - $\rightarrow$ Otherwise, data leakage $\Rightarrow$ Optimistic estimates

- **Subsets** of the assortment: $\mathcal{D} = \underbrace{\mathcal{D}_{train}}_{Train} \cup \underbrace{\mathcal{D}_{validation}}_{Tune} \cup \underbrace{\mathcal{D}_{test}}_{Assess}$
  - $\rightarrow$ Disjoint subsets: $\mathcal{D}_{train} \cap \mathcal{D}_{validation} \cap \mathcal{D}_{test} = \emptyset$

- **Considerations** (typical):

# Formalization

- **Premise**: The procedures must use different data partitions
  - $\rightarrow$ Otherwise, data leakage $\Rightarrow$ Optimistic estimates

- **Subsets** of the assortment: $\mathcal{D} = \underbrace{\mathcal{D}_{train}}_{Train} \cup \underbrace{\mathcal{D}_{validation}}_{Tune} \cup \underbrace{\mathcal{D}_{test}}_{Assess}$
  - $\rightarrow$ Disjoint subsets: $\mathcal{D}_{train} \cap \mathcal{D}_{validation} \cap \mathcal{D}_{test} = \emptyset$

- **Considerations** (typical):
  1. $\mathcal{D}_{train}$ is the largest subset: $|\mathcal{D}_{train}| \gg |\mathcal{D}_{validation}|, |\mathcal{D}_{test}|$

# Formalization

- **Premise**: The procedures must use different data partitions
  - $\rightarrow$ Otherwise, data leakage $\Rightarrow$ Optimistic estimates

- **Subsets** of the assortment: $\mathcal{D} = \underbrace{\mathcal{D}_{train}}_{Train} \cup \underbrace{\mathcal{D}_{validation}}_{Tune} \cup \underbrace{\mathcal{D}_{test}}_{Assess}$
  - $\rightarrow$ Disjoint subsets: $\mathcal{D}_{train} \cap \mathcal{D}_{validation} \cap \mathcal{D}_{test} = \emptyset$

- **Considerations** (typical):
  1. $\mathcal{D}_{train}$ is the largest subset: $|\mathcal{D}_{train}| \gg |\mathcal{D}_{validation}|, |\mathcal{D}_{test}|$
  2. $\mathcal{D}_{validation}$ is a subset of $\mathcal{D}_{train}$: $\mathcal{D}_{validation} \subset \mathcal{D}_{train}$

# Formalization

- **Premise**: The procedures must use different data partitions
  - $\rightarrow$ Otherwise, data leakage $\Rightarrow$ Optimistic estimates

- **Subsets** of the assortment: $\mathcal{D} = \underbrace{\mathcal{D}_{train}}_{Train} \cup \underbrace{\mathcal{D}_{validation}}_{Tune} \cup \underbrace{\mathcal{D}_{test}}_{Assess}$
  - $\rightarrow$ Disjoint subsets: $\mathcal{D}_{train} \cap \mathcal{D}_{validation} \cap \mathcal{D}_{test} = \emptyset$

- **Considerations** (typical):
  1. $\mathcal{D}_{train}$ is the largest subset: $|\mathcal{D}_{train}| \gg |\mathcal{D}_{validation}|, |\mathcal{D}_{test}|$
  2. $\mathcal{D}_{validation}$ is a subset of $\mathcal{D}_{train}$: $\mathcal{D}_{validation} \subset \mathcal{D}_{train}$
  3. $\mathcal{D}_{test}$ may be external

# Formalization

- **Premise**: The procedures must use different data partitions
  - $\rightarrow$ Otherwise, data leakage $\Rightarrow$ Optimistic estimates

- **Subsets** of the assortment: $\mathcal{D} = \underbrace{\mathcal{D}_{train}}_{Train} \cup \underbrace{\mathcal{D}_{validation}}_{Tune} \cup \underbrace{\mathcal{D}_{test}}_{Assess}$
  - $\rightarrow$ Disjoint subsets: $\mathcal{D}_{train} \cap \mathcal{D}_{validation} \cap \mathcal{D}_{test} = \emptyset$

- **Considerations** (typical):
  1. $\mathcal{D}_{train}$ is the largest subset: $|\mathcal{D}_{train}| \gg |\mathcal{D}_{validation}|, |\mathcal{D}_{test}|$
  2. $\mathcal{D}_{validation}$ is a subset of $\mathcal{D}_{train}$: $\mathcal{D}_{validation} \subset \mathcal{D}_{train}$
  3. $\mathcal{D}_{test}$ may be external
  4. In general, random division
     - $\rightarrow$ Stratification to maintain label distribution

# Summary

| **Partition** | **Purpose** | **Used for** | **Ratio** (%) |
|---|---|---|---|
| $D_{train}$ | Fit model parameters | Learning the model | $60 - 80$ |
| $D_{validation}$ | Tune hyperparameters, model selection | Model selection | $10 - 20$ |
| $D_{test}$ | Evaluate final performance on unseen data | Assessment | $10 - 20$ |

# Introduction

- So far $\Rightarrow$ static partitions over assortment $\mathcal{D}$

# Introduction

- So far $\Rightarrow$ static partitions over assortment $\mathcal{D}$
  - $\rightarrow$ Possible biases due to skews in $\mathcal{D}$

# Introduction

- So far $\Rightarrow$ static partitions over assortment $\mathcal{D}$
  - $\rightarrow$ Possible biases due to skews in $\mathcal{D}$
- **Possible solution**: *Repeatedly* perform data *partitioning* on $\mathcal{D}$

# Introduction

- So far $\Rightarrow$ static partitions over assortment $\mathcal{D}$
  - $\rightarrow$ Possible biases due to skews in $\mathcal{D}$
- **Possible solution**: *Repeatedly* perform data *partitioning* on $\mathcal{D}$
  - $\rightarrow$ Reduces variance of the performance estimate (by averaging)
  - $\rightarrow$ Reduces estimation bias compared to a single partitioning

# Introduction

- So far $\Rightarrow$ static partitions over assortment $\mathcal{D}$
  - $\rightarrow$ Possible biases due to skews in $\mathcal{D}$
- **Possible solution**: *Repeatedly* perform data *partitioning* on $\mathcal{D}$
  - $\rightarrow$ Reduces variance of the performance estimate (by averaging)
  - $\rightarrow$ Reduces estimation bias compared to a single partitioning
- Typical strategies:
  1. Hold-out validation
  2. $k$-fold Cross-validation
  3. Stratified $k$-fold Cross-validation
  4. Leave-one-out Cross-validation
  5. Leave-p-out Cross-validation

# Strategies

1. Hold-out validation
   - Simplest case: fixed $\mathcal{D}_{train}$, $\mathcal{D}_{validation}$, and $\mathcal{D}_{test}$
   - Error is computed only once
   - Features:
     - ✓ Fast
     - ✗ High variance (split-dependent)

# Strategies

1. Hold-out validation
   - Simplest case: fixed $\mathcal{D}_{train}$, $\mathcal{D}_{validation}$, and $\mathcal{D}_{test}$
   - Error is computed only once
   - Features:
     - ✓ Fast
     - ✗ High variance (split-dependent)

2. $k$-fold CV
   - Split data into $k$ roughly equal folds
   - Train with $k - 1$ partitions; test with the remaining one
   - Repeat the procedure $k$ times $\Rightarrow$ As many as folds created
   - Features:
     - ✗ **Bias**: Slightly over-optimistic for small $k$ ($\mathcal{D}_{train}$ sets are smaller)
     - ✓ **Variance**: Decreases as $k$ decreases

# Strategies

3. Stratified $k$-fold CV
   - Variant of $k$-fold where class proportions are preserved in each fold (label imbalance)

# Strategies

3. Stratified $k$-fold CV
   - Variant of $k$-fold where class proportions are preserved in each fold (label imbalance)
4. Leave-one-out CV
   - Extreme case of $k$-fold CV with $k = |\mathcal{D}|$
   - Train with $|\mathcal{D}| - 1$ points; test with the remaining one
   - Features:
     - ✓ **Bias**: Very low (trained on nearly full dataset)
     - ✗ **Variance**: Very high (small changes in data affect each fold)

# Strategies

3. Stratified $k$-fold CV
   - Variant of $k$-fold where class proportions are preserved in each fold (label imbalance)
4. Leave-one-out CV
   - Extreme case of $k$-fold CV with $k = |\mathcal{D}|$
   - Train with $|\mathcal{D}| - 1$ points; test with the remaining one
   - Features:
     - ✓ **Bias**: Very low (trained on nearly full dataset)
     - ✗ **Variance**: Very high (small changes in data affect each fold)
5. Leave-$p$-out CV
   - Generalization of leave-one-out: leaves $p$ samples out at a time
   - Theoretical number of folds: $\binom{|\mathcal{D}|}{p}$
     - $\rightarrow$ Infeasible in practical cases unless $p$ is low
   - Used mainly in theoretical studies of CV properties

# Summary

| Strategy | Bias | Variance | Cost |
|---|---|---|---|
| Hold-out | Medium | High | 1 fit |
| (Stratified) $k$-fold | Low | Medium | $k$ fits |
| Leave-one-out | Very low | High | $\|\mathcal{D}\|$ fits |
| Leave-$p$-out | Very low | Very high | $\binom{\|\mathcal{D}\|}{p}$ fits |