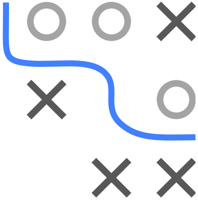
Introduction to Machine Learning

Hyperparameter Tuning
In a Nutshell

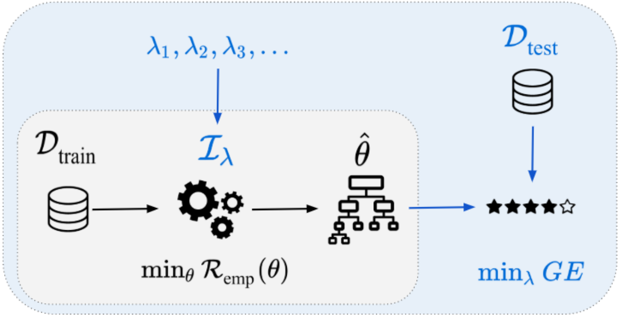
Learning goals

- Understand the main idea behind tuning,
- fulfilling the untouched-test set principle via nested resampling,
- and pipelines

WHAT IS TUNING?

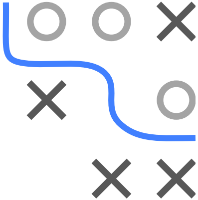


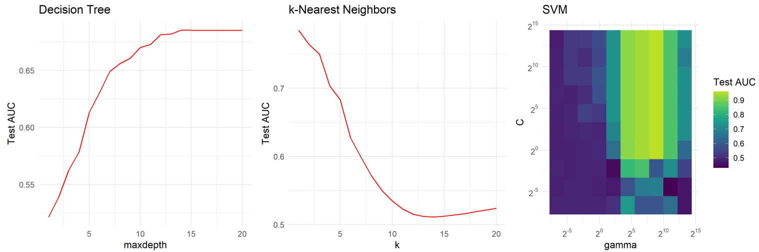
- Tuning is the process of selecting the best hyperparameters, denoted as λ, for a machine learning model.
- Hyperparameters are the parameters of the learner (versus model parameters θ).
- Consider a guitar analogy: Hyperparameters are akin to the tuning pegs. Learning the best parameters $\hat{\theta}$ – playing the guitar – is a separate process that depends on tuning.



WHY TUNING MATTERS

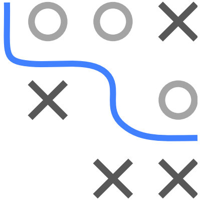
- Just like a guitar won't perform well when out-of-tune, properly tuning a learner can drastically improve the resulting model performance.
- Tuning helps find a balance between underfitting and overfitting.





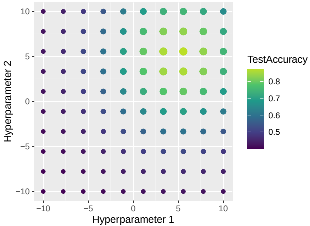
Comparing AUCs of different values for hyperparameters *maxdepth, k, gamma, and C*

HOW HARD COULD IT BE?



- Very difficult: There are lots of different configurations to choose from, known as the hyperparameter space, denoted by $\mathbf{\Lambda}$ (analogous to Θ).
- Black box: If one opts for a configuration $\boldsymbol{\lambda} \in \mathbf{\Lambda}$, how can its performance be measured (and compared)?

$\Rightarrow$ Well-thought-out **Black-Box Optimization Techniques** are needed.



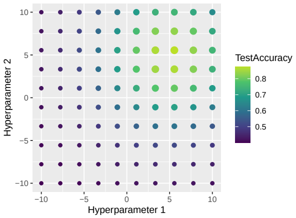
Exponential growth of $\mathbf{\Lambda}$: For two discrete hyperparameters with each 10 possible values, $10 \cdot 10 = 100$ configurations can be evaluated

NAÏVE APPROACHES

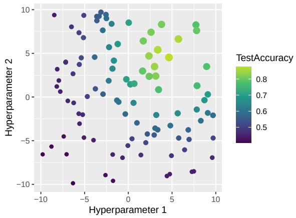
Goal: Find a best configuration $\boldsymbol{\lambda}^* \in \arg\min_{\boldsymbol{\lambda} \in \boldsymbol{\Lambda}} \widehat{\mathrm{GE}}(\mathcal{I}, \rho, \boldsymbol{\lambda})$

$\Rightarrow$ Tuners τ, e.g., **Grid Search** and **Random Search**, output a $\boldsymbol{\lambda}^*$

Grid Search

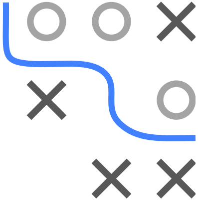


Random Search



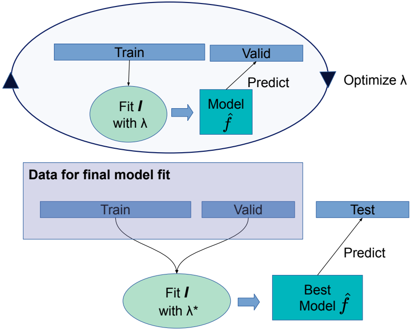
Sophisticated techniques, based on assumptions about the objective function, search for optimal solutions more efficiently.

UNTOUCHED-TEST-SET PRINCIPLE



We've found a $\boldsymbol{\lambda}^* \in \boldsymbol{\Lambda}$. How well does it perform?

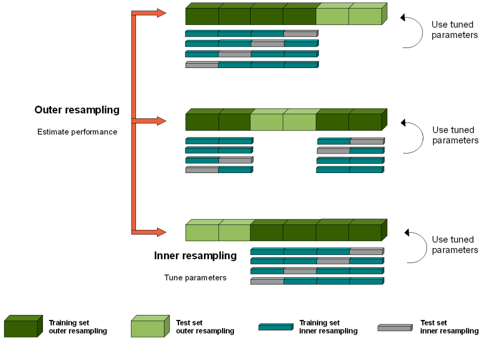
- **Careful:** We cannot use the same data for both tuning and performance estimation, as this would lead to (optimistically) biased performance estimates!
- To obtain an unbiased $\widehat{\mathrm{GE}}$, we need an **untouched** test set:



NESTED RESAMPLING

To decrease variance of the $\widehat{\mathrm{GE}}$, **Nested Resampling** is used:

- Just as we generalized holdout splitting to resampling, we generalize the three-way split to nested resampling (as we first have to find $\boldsymbol{\lambda}^*$):

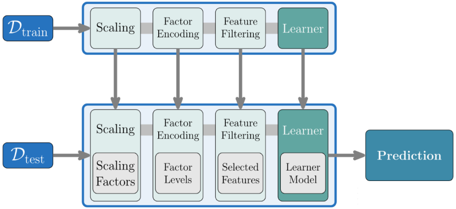


PIPELINES IN MACHINE LEARNING

Pipelines are like the assembly lines in machine learning. They automate the sequence of data processing and model building tasks.

Why Pipelines Matter:

- **Streamlined Workflow:** Automates the flow from data preprocessing to model training.
- **Reproducibility:** Ensures that results can be reproduced consistently.
- **Error Reduction:** Minimizes the chance of human errors in the model building process.



©