# **Introduction to Machine Learning**

## **Tuning: In a Nutshell**



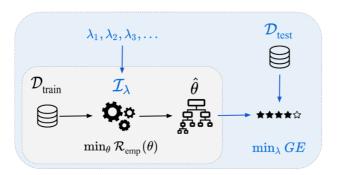
### Learning goals

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



#### WHAT IS TUNING?

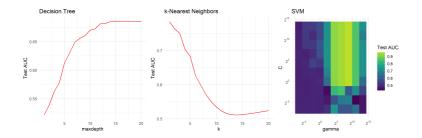
- Tuning is the process of selecting the best hyperparameters, denoted as  $\lambda$ , for a machine learning model.
- Hyperparameters are the parameters of the learner (versus model parameters  $\theta$ ).
- Consider a guitar analogy: Hyperparameters are akin to the tuning pegs. Learning the best parameters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



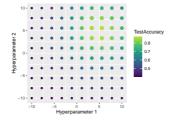
× CO × X

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

### **HOW HARD COULD IT BE?**

- ullet Very difficult: There are lots of different configurations to choose from, known as the hyperparameter space, denoted by  $\Lambda$  (analogous to  $\Theta$ ).
- Black box: If one opts for a configuration  $\lambda \in \Lambda$ , how can its performance be measured (and compared)?
- ⇒ Well-thought-out Black-Box Optimization Techniques are needed.



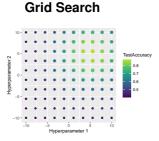


Exponential growth of  $\Lambda$ : For two discrete hyperparameters with each 10 possible values,  $10\cdot 10=100$  configurations can be evaluated

### **NAÏVE APPROACHES**

Goal: Find a best configuration  $\lambda^* \in \arg\min_{\lambda \in \Lambda} \widehat{\mathrm{GE}}(\mathcal{I}, \rho, \lambda)$  $\Rightarrow$  Tuners  $\tau$ , e.g., **Grid Search** and **Random Search**, output a  $\lambda^*$ 





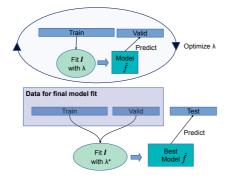


Sophisticated techniques, based on assumptions about the objective function, search for optimal solutions more efficiently.

### **UNTOUCHED-TEST-SET PRINCIPLE**

We've found a  $\lambda^* \in \Lambda$ . 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!
- $\bullet$  To obtain an unbiased  $\widehat{GE},$  we need an untouched test set:

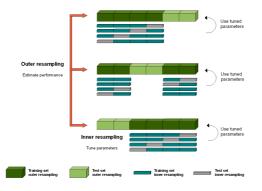




### **NESTED RESAMPLING**

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

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





### 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.



