Hyperparameter optimization

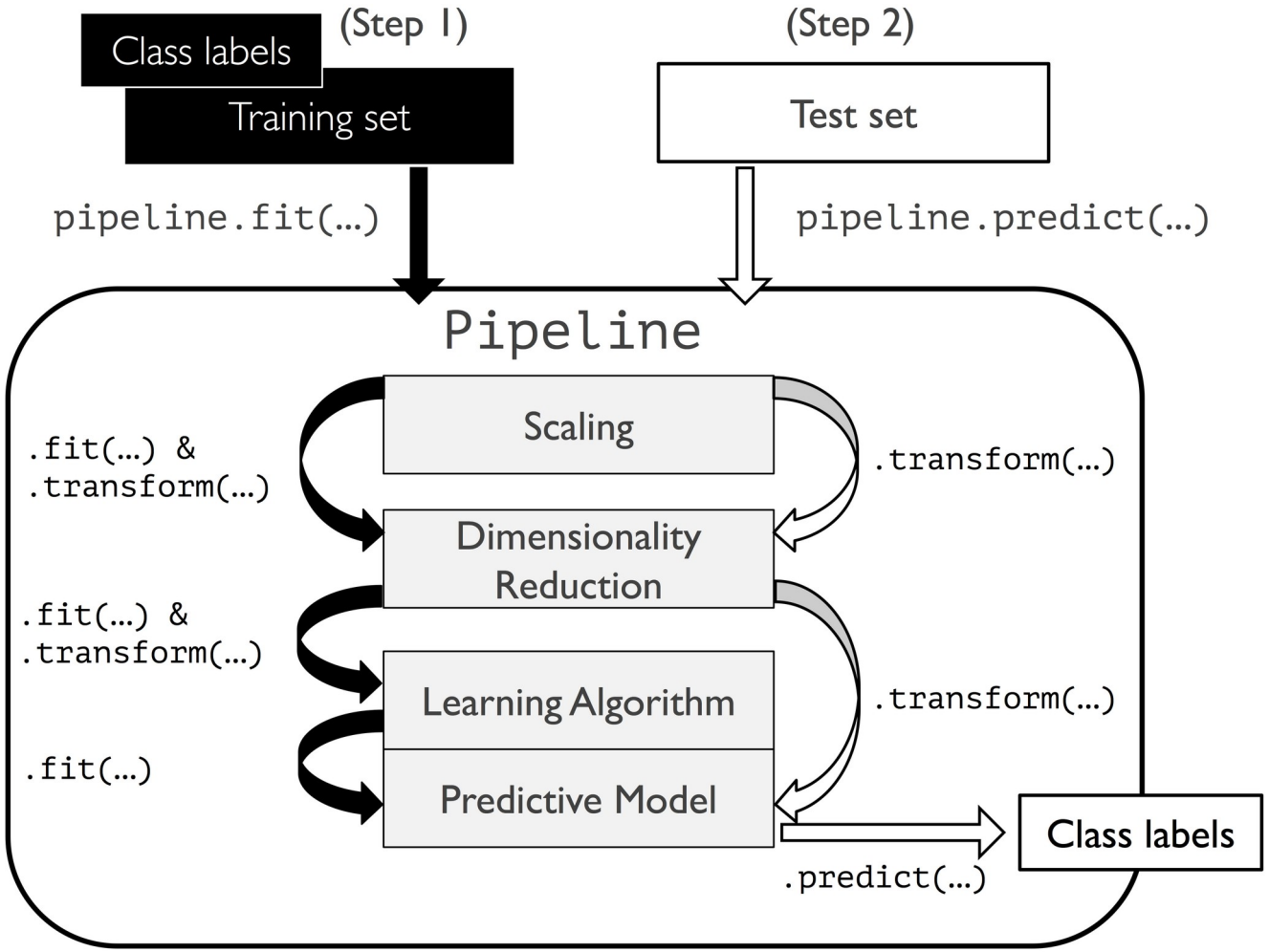
Summary up until evaluation metrics

see Ch. 06 in book “Python Machine Learning” by Raschka & Mirjalili

Overview

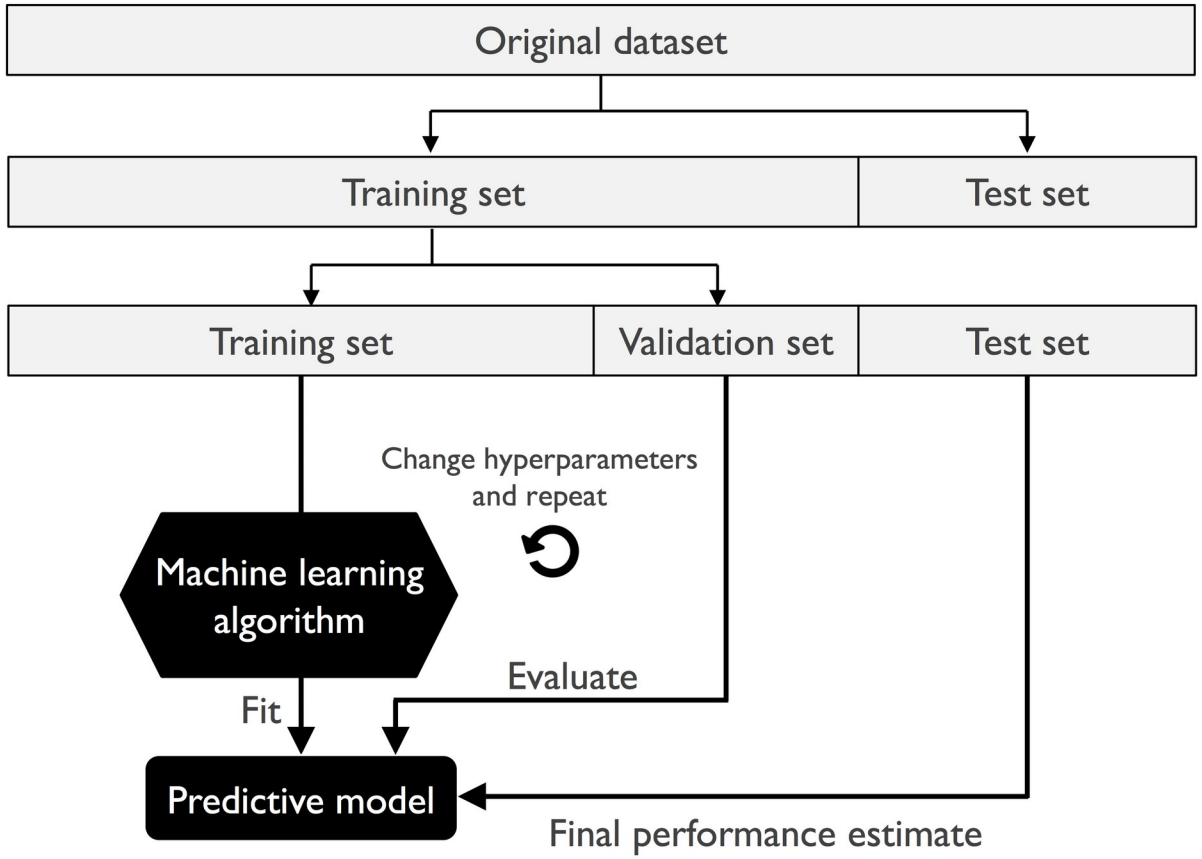
- Pipelines
- Holdout cross-validation
- K-fold cross-validation
- Learning & validation curves
- Grid search & randomized search
- Nested cross-validation
- Evaluation metrics (not included in this presentation)

Pipelines


Class labels
(Step 1)
Training set
(Step 2)
Test set
pipeline.fit(...)
pipeline.predict(...)
Pipeline
Scaling
.fit(...) &
.transform(...)
.transform(...)
Dimensionality
Reduction
.fit(...) &
.transform(...)
Learning Algorithm
.transform(...)
.fit(...)
Predictive Model
Class labels
.predict(...)

The holdout method (“validation” partition)

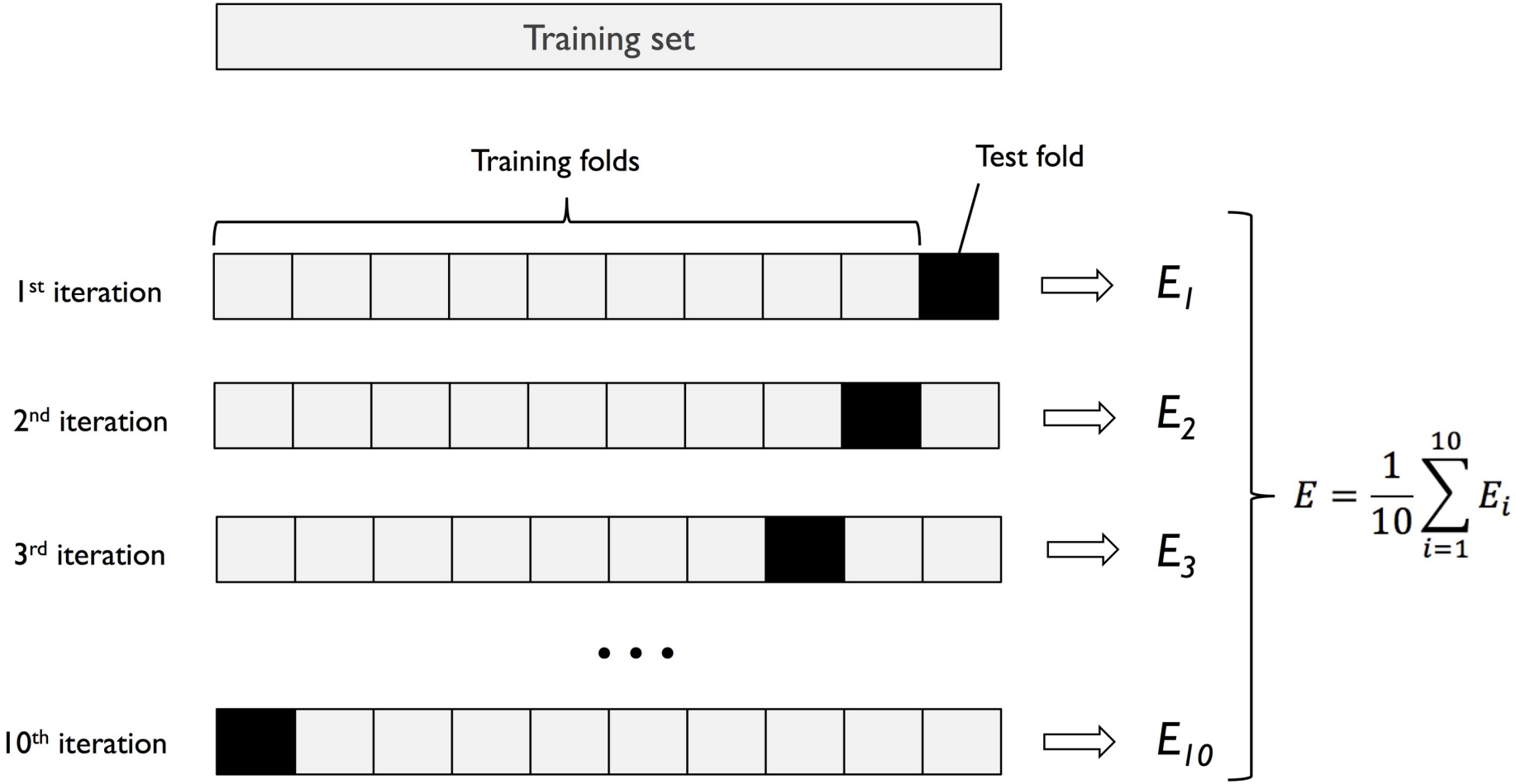
- The method of splitting the full dataset into training partition and an evaluation partition is referred to as the *hold out method*



Stop here for holdout validation

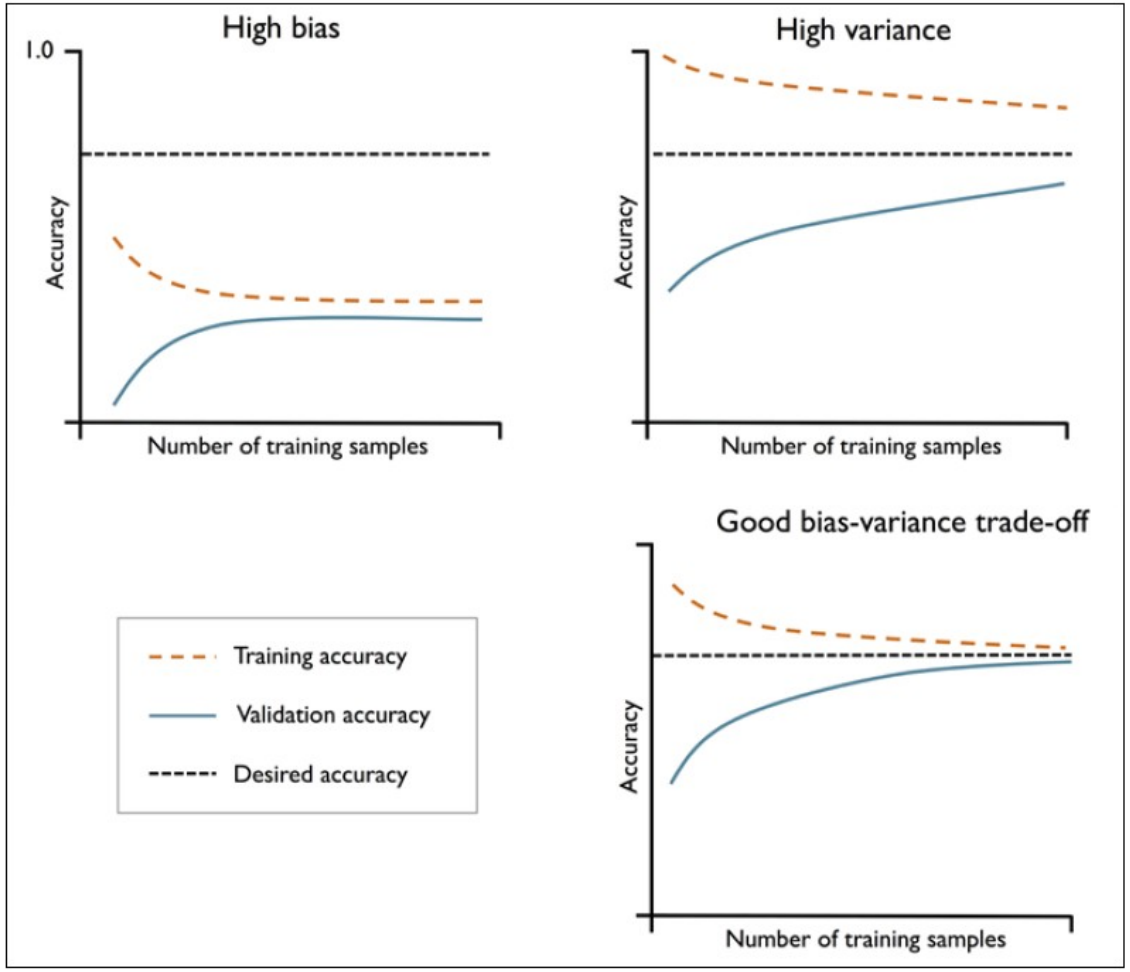
When splitting training set again it’s called **holdout cross-validation**

K-fold cross-validation (CV)



Learning & validation curves

Underfitting, try:
- Construct or collect more features
- More flexible modelling
- Less regularisation

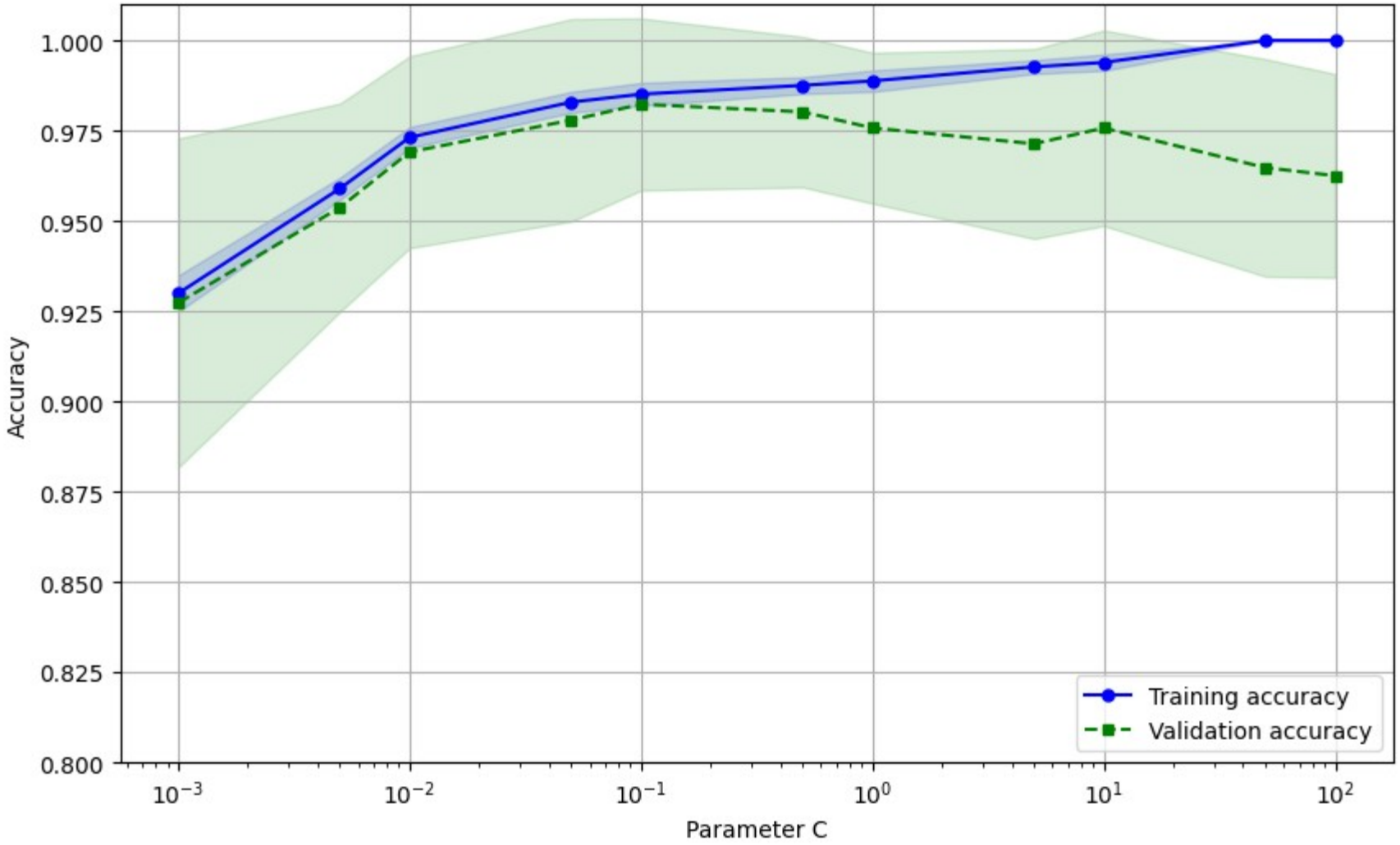


Overfitting, try:
- More data / fewer features
- Less flexible modelling
- More regularization

Balanced fitting, try:
- Pat yourself on the back
- Perform victory dance
- Brag about it

Learning & validation curves

Validation curve for SVC
- Here we varied the C parameter
- For each C, 10-fold CV was performed
- Plot two lines of training and validation accuracy

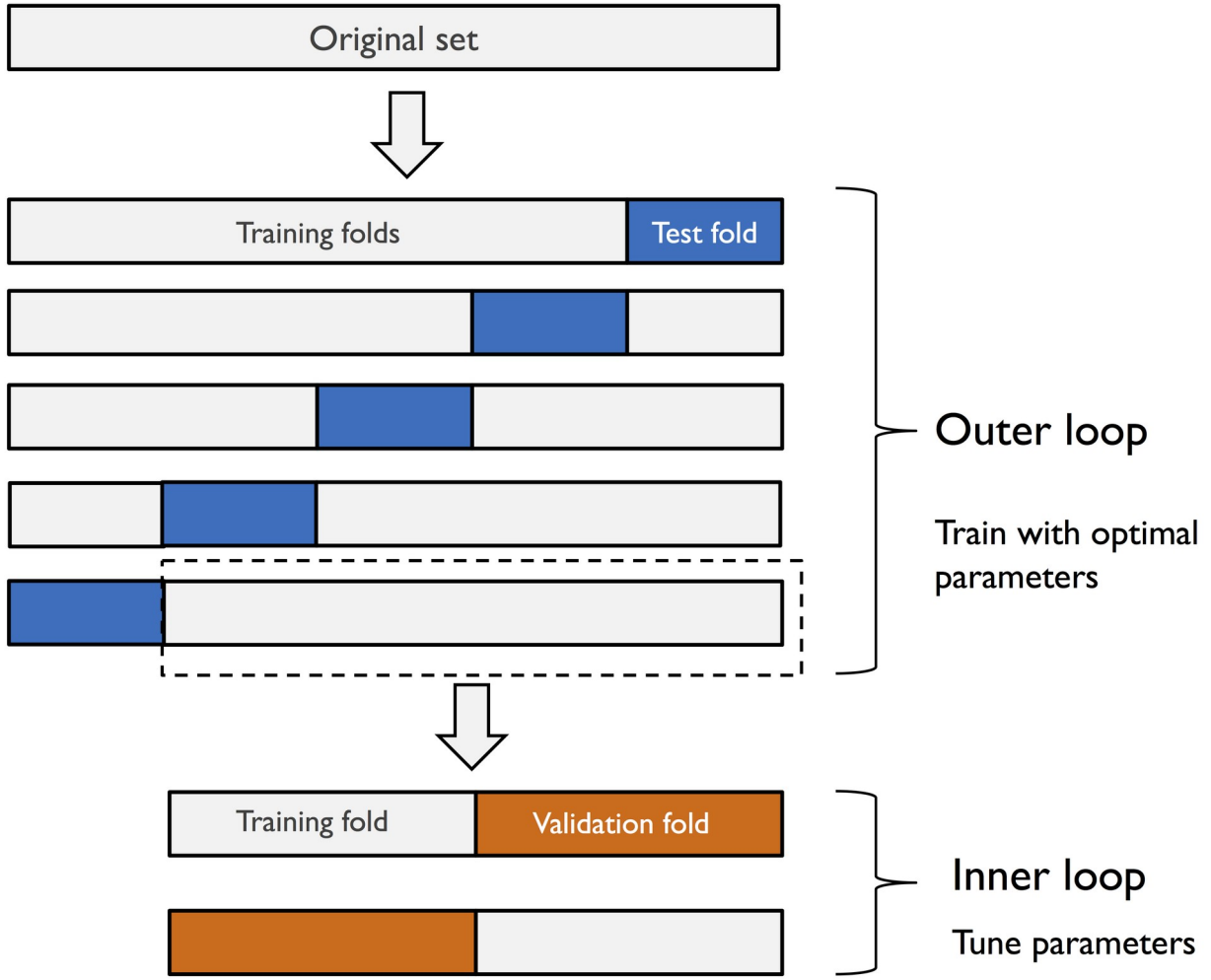


Grid search & randomized search

- Grid search:
  - Brute-force exhaustive search through grid of specified set of hyperparameters
- Randomized search:
  - We **don’t** specify a grid of hyperparameter combinations to search exhaustively
  - Instead we specify
    - A range of possible hyperparameter values (could be continous)
    - Parameter specific probability distributions
    - Max set of iterations

Nested cross-validation

- Cross-validation loop within a cross-validation loop
- Addresses the fact that the initial split between the training/val set and test set is also sensitive to how the split is done
- Becomes very computationally expensive
- Is rarely done when you are working with datasets of over a certain size



Thank you for listening