

# Feature Engineering, Model Selection and Pipelining



# Feature Engineering

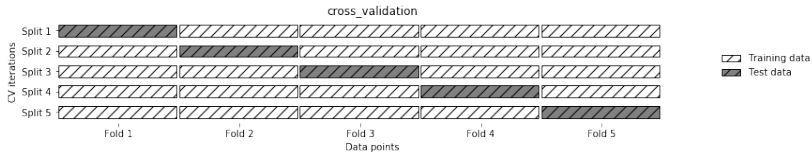
1. Categories into dummies (One-Hot-Encoding)
2. Continuous variables into dummies representing groups (Binning and Discretization)
3. Polynomials
4. Combinations

# Cross-Validation

- ▶ Learned weights likely specific to training set
- ▶ Estimates of generalization affected by random split into training and test
- ▶ Solution: Repeat learning on different splits

! **!! ALWAYS !!** use this

# Cross-Validation, cont.



from: Andreas Müller and Sarah Guido (2016): Introduction to Machine Learning with Python, O'Reilly

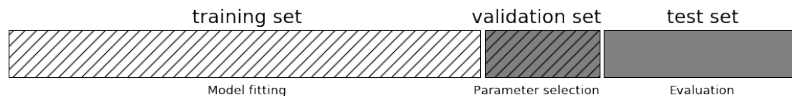
## Cross-Validation strategies

- ▶ k-Fold CV: Split evenly into  $k$  data points, pick one as test set and the rest as training set, repeat  $k$  times (data can be shuffled first)
- ▶ Stratified k-Fold: Split data  $k$  times such that proportions between classes are similar across folds
- ▶ Leave-one-out CV: Set  $K$  equal to the number of observations
- ▶ Shuffle-split CV: In each fold, split data into fixed shares for training and test set (which do not need add up to 1)

# Grid Search

- ▶ Loop over different combinations of parameters
- ▶ Keep the best performing parameter combinations returning
- ▶ IMPORTANT: Don't evaluate parameters on training set, but on distinct *validation set*

# Validation set



from: Andreas Müller and Sarah Guido (2016): Introduction to Machine Learning with Python, O'Reilly

- ▶ Necessary to evaluate parameter combinations on unseen data
- ▶ ... for the same reason you do generalize on unseen data, too
- ▶ Simply split training set again

# Grid Search with Cross Validation

- ▶ `GridSearchCV(estimator, param_grid)` (→ [Documentation](#))
  - ▶ `estimator` is model class (i.e. `MLPerceptron()`)
  - ▶ `param_grid` is dict or list of dict
  - ▶ Optionally specify desired evaluation score and CV strategy



# Grid Search with Cross Validation

- ▶ `GridSearchCV(estimator, param_grid)` (→ [Documentation](#))
  - ▶ `estimator` is model class (i.e. `MLPerceptron()`)
  - ▶ `param_grid` is dict or list of dict
  - ▶ Optionally specify desired evaluation score and CV strategy
- ▶ How many computations do you have for a 5-fold Cross-Validation, 2 possibilities for one parameter and 3 for another parameters?

# Model Pipelines

What's wrong about scaling, then folding and then selecting parameters?

# Model Pipelines

What's wrong about scaling, then folding and then selecting parameters?

- ▶ The information used for scaling partly comes from the verification fold
- ▶ This is not how new data looks to the model
- ▶ → Information leakage

# Model Pipelines

What's wrong about scaling, then folding and then selecting parameters?

- ▶ The information used for scaling partly comes from the verification fold
- ▶ This is not how new data looks to the model
- ▶ → Information leakage Right approach: Splitting/Folding before any preprocessing, i.e. in the cross-validation loop
- ▶ Pipeline() to the rescue (→ [Documentation](#))

# Checklist

- ▶ Learn any model *only* using GridSearchCV
- ▶ Put parameters into dictionary
- ▶ If you scale data, you *must* use Pipeline()