Machine Learning & Pattern Recognition

SONG Xuemeng

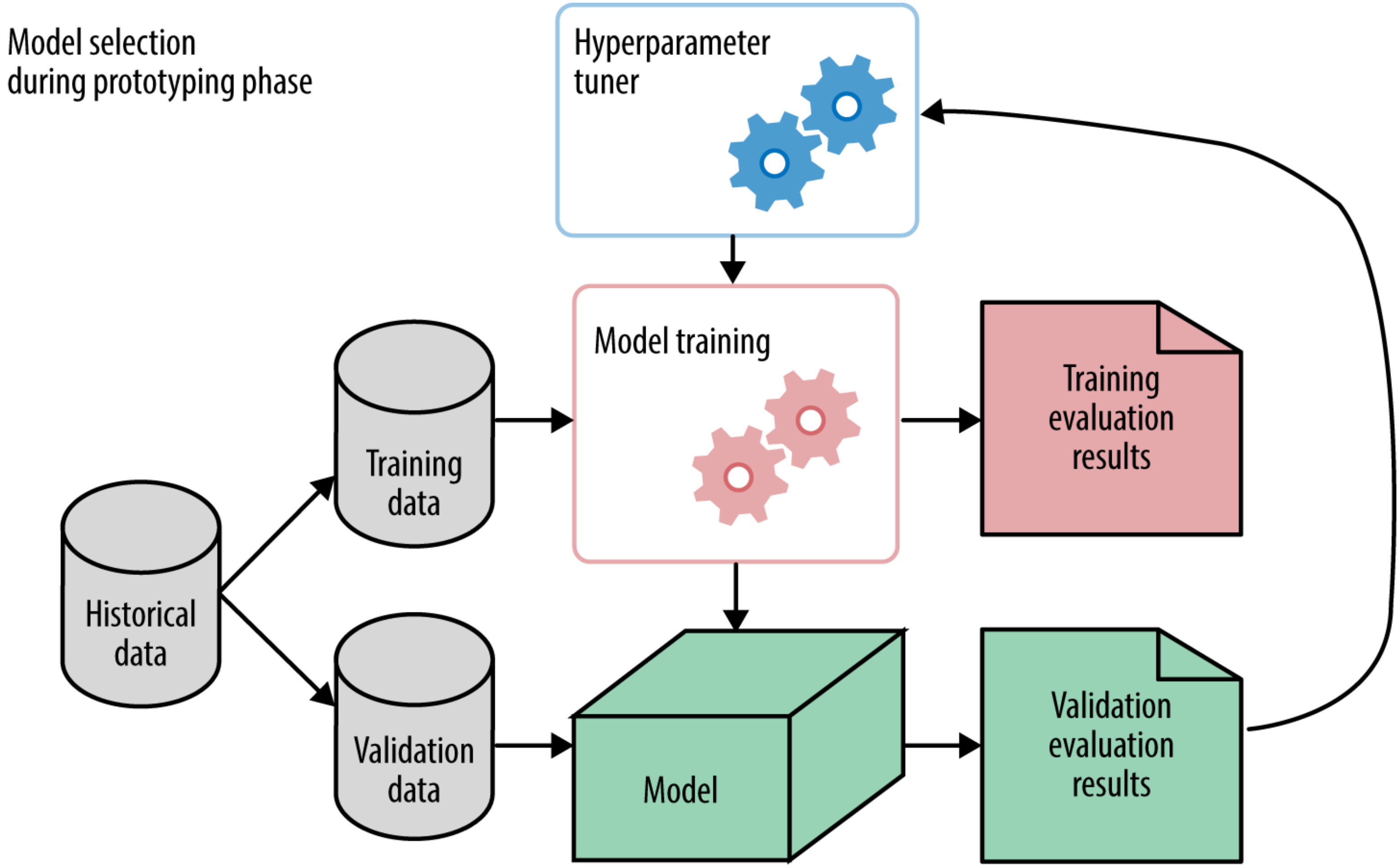
sxmustc@gmail.com

http://xuemeng.bitcron.com/

Model Selection

Model Selection

Model selection refers to the process of selecting the right model (or type of model) that fits the data.



Model Parameters Versus Hyperparameters

$\mathbf{w}$: *model parameter* is learned during the training phase.
λ: *hyperparameters* are values that must be specified outside of the training procedure.

Linear regression: $$\min_{\mathbf{w}} E(\mathbf{w}) = \frac{1}{2N}\|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2$$

Ridge regression: $$\min_{\mathbf{w}} E(\mathbf{w}) = \frac{1}{2N}\|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$$

Lasso: $$\min_{\mathbf{w}} E(\mathbf{w}) = \frac{1}{2N}\|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 + \frac{\lambda}{2}\|\mathbf{w}\|_1$$

- Decision trees → the desired depth and number of leaves.
- (SVM)→ a misclassification penalty term.
- Kernelized SVMs → kernel parameters like the width for radial basis function (RBF) kernels.
- …

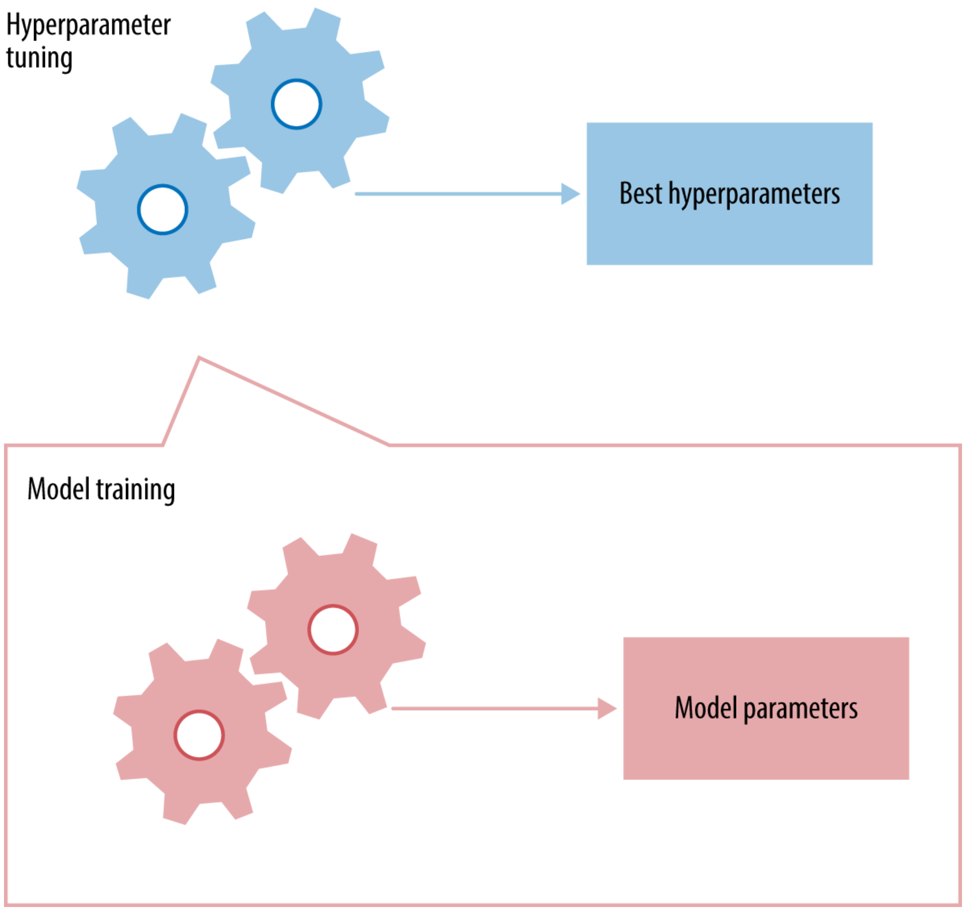
What Do Hyperparameters Do?

- A regularization hyperparameter controls the *capacity* of the model.
- Proper control of model capacity can prevent overfitting.

- Another type of hyperparameter comes from the training process itself.
- For instance, stochastic gradient descent (SGD) optimization requires a learning rate, batch size.
- Some optimization methods require a convergence threshold.
- These also need to be set to reasonable values in order for the training process to find a good model.

Hyperparameter Tuning

Each trial of a particular hyperparameter setting involves training a model—an inner optimization process.



The outcome of hyperparameter tuning is the best hyperparameter setting.

The outcome of model training is the best model parameter setting.

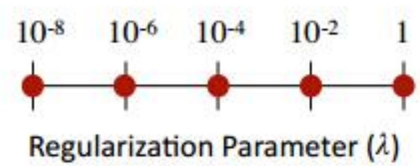
Pseudo-Python Code

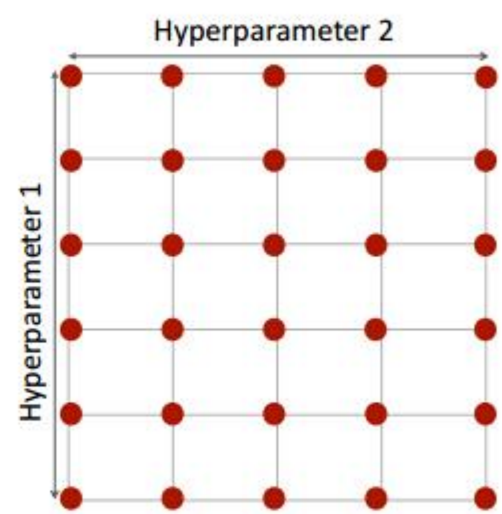
```
func hyperparameter_tuner (training_data,
                           validation_data,
                           hp_list):
    hp_perf = []

    # train and evaluate on all hyperparameter settings
    foreach hp_setting in hp_list:
        m = train_model(training_data, hp_setting)
        validation_results = eval_model(m, validation_data)
        hp_perf.append(validation_results)

    # find the best hyperparameter setting
    best_hp_setting = hp_list[max_index(hp_perf)]

    # IMPORTANT:
    # train a model on *all* available data using the best
    # hyperparameters
    best_m = train_model(training_data.append(validation_data),
                         best_hp_setting)

    return (best_hp_setting, best_m)
```

Hyperparameter Turning Algorithm-Grid Search

Grid search picks out a grid of hyperparameter values, evaluates every one of them, and returns the winner.

- If the hyperparameter is the number of leaves in a decision tree, then the grid could be 10, 20, 30, …, 100.
- For regularization parameters, it is common to use exponential scale: 1e-5, 1e-4, 1e-3, …, 1.



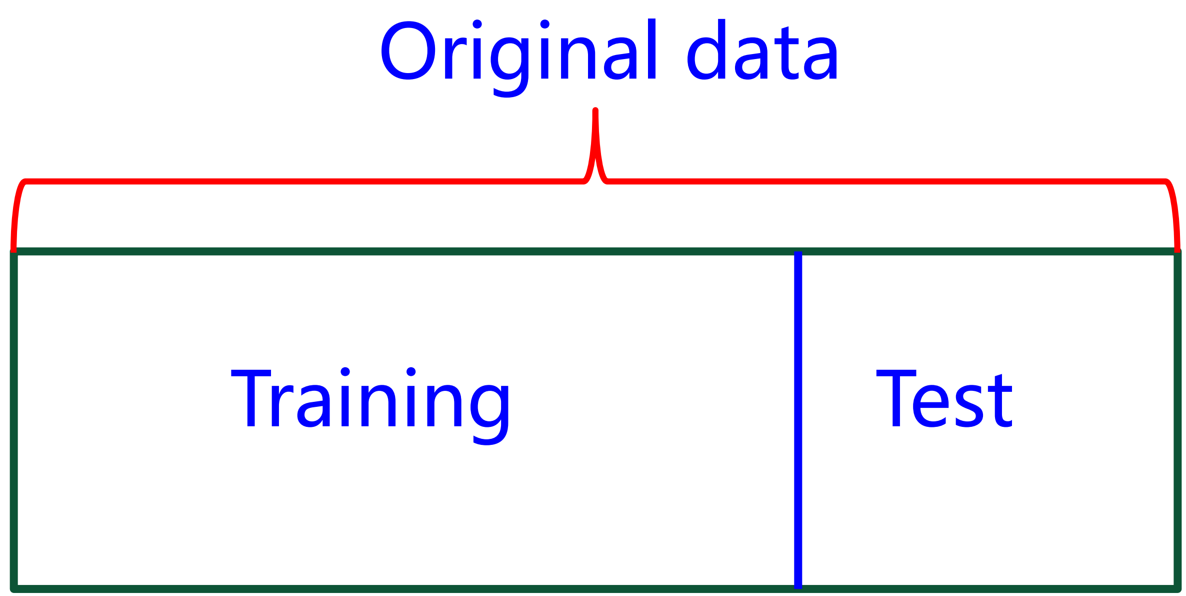


Evaluation the Performance of a Classifier

- Holdout Method
- Random Subsampling
- Cross-validation
- Bootstrap

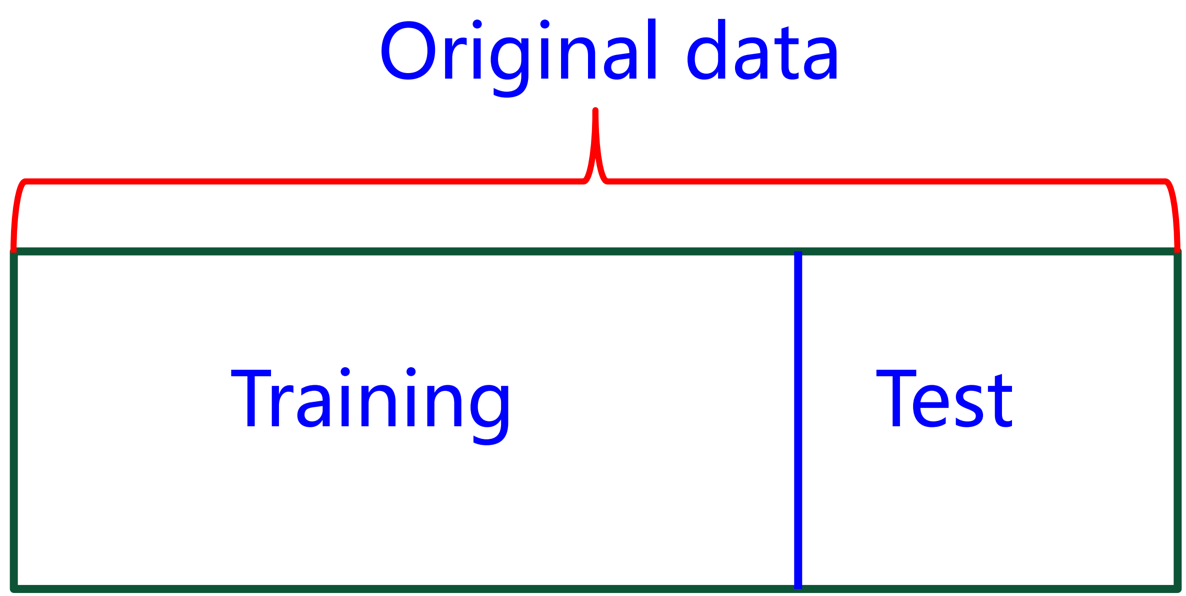
Holdout Method

- The original data with labeled examples is partitioned into two disjoint sets, called the training and test sets, respectively.
- The model is induced from the training set.
- The performance is evaluated on the test set.



Holdout Method

- The proportion of data reserved for training and for testing is typically at the discretion of the analysts (e.g., 1/2-1/2 or 2/3 for training and 1/3 for testing).
- **Limitations:**
  - Fewer labeled examples are available for training because some are withheld for testing.



Random Subsampling

- The holdout method can be repeated several times to improve the performance.
- Let acc_i be the model accuracy during the i^{th} iteration. The overall accuracy is given by $acc_{sub} = \sum_{i=1}^{k} acc_i \,/k$.

- **Limitations:**
  - Random subsampling also does not utilize as much data as possible for training.
  - It also has no control over the number of times each record is used for testing and training.

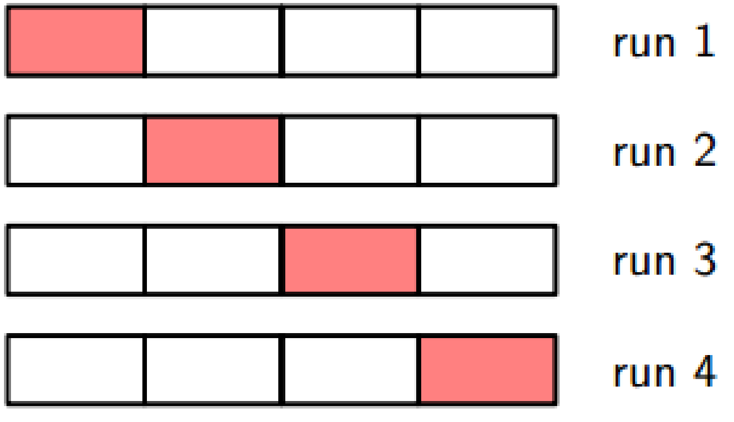
Cross-validation

- Suppose we partition the data into two equal-sized subsets.
- We choose one of the subsets for training and the other for testing.
- We then swap the roles of the subsets so that the previous training set becomes the test set and vice versa.
- The total error is obtained by averaging the errors for both runs.

Two-fold cross-validation.

Cross-validation

- **More general:** the k-fold cross-validation
- Segments the data into k equal-sized partitions.
- During each run, one partition is chosen for testing, while the rest are used for training.
- This procedure is repeated k times so that each partition is used for testing exactly once.
- The total error is found by averaging the errors for all k runs.

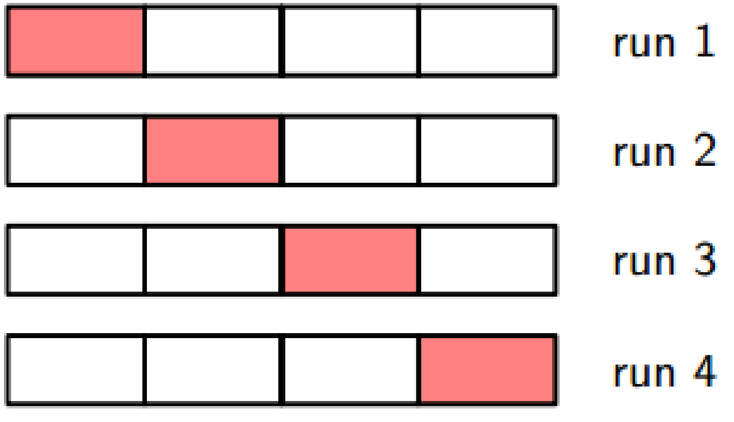


The technique of *k*-fold cross-validation, illustrated here for the case of k= 4.

Cross-validation

- A special case of the k-fold cross-validation sets k=N, the size of the dataset. This is the so-called *leave-one-out* approach, each test set contains only one record.

- The methods presented so far assume that the training records are sampled without replacement.
- There are no duplicate records in the training and test sets.



The technique of *k*-fold cross-validation, illustrated here for the case of k= 4.

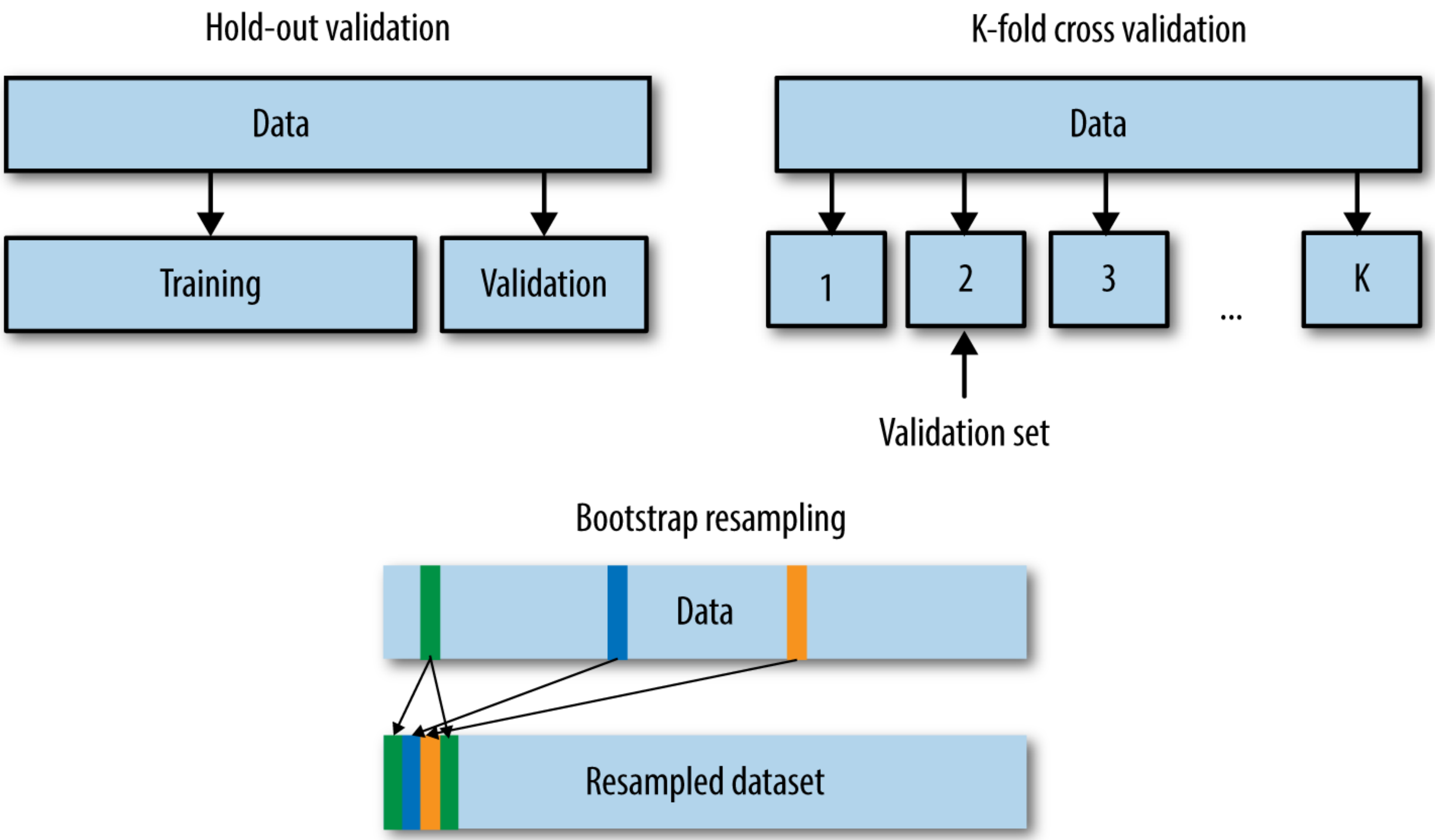
Bootstrap

- In bootstrap, the training records are sampled with replacement, i.e., a record already chosen for training is put back into the original pool of records so that it is equally likely to be redrawn.
- If the original data has N records, on average, a bootstrap sample of size N contains about 63.2% of the records in the original data.
  - The probability a record is chosen by a bootstrap sample is

$$1-\left(1-\frac{1}{N}\right)^{N} \rightarrow 1-e^{-1}=0.632$$

- Records that are not included in the bootstrap sample become part of the test set.
- The sampling procedure is then repeated b times to generate b bootstrap samples.

Comparison


Hold-out validation
Data
Training
Validation
K-fold cross validation
Data
1
2
3
...
K
Validation set
Bootstrap resampling
Data
Resampled dataset