Evaluating Performance II

Lecture 07

Spot the misstep

1

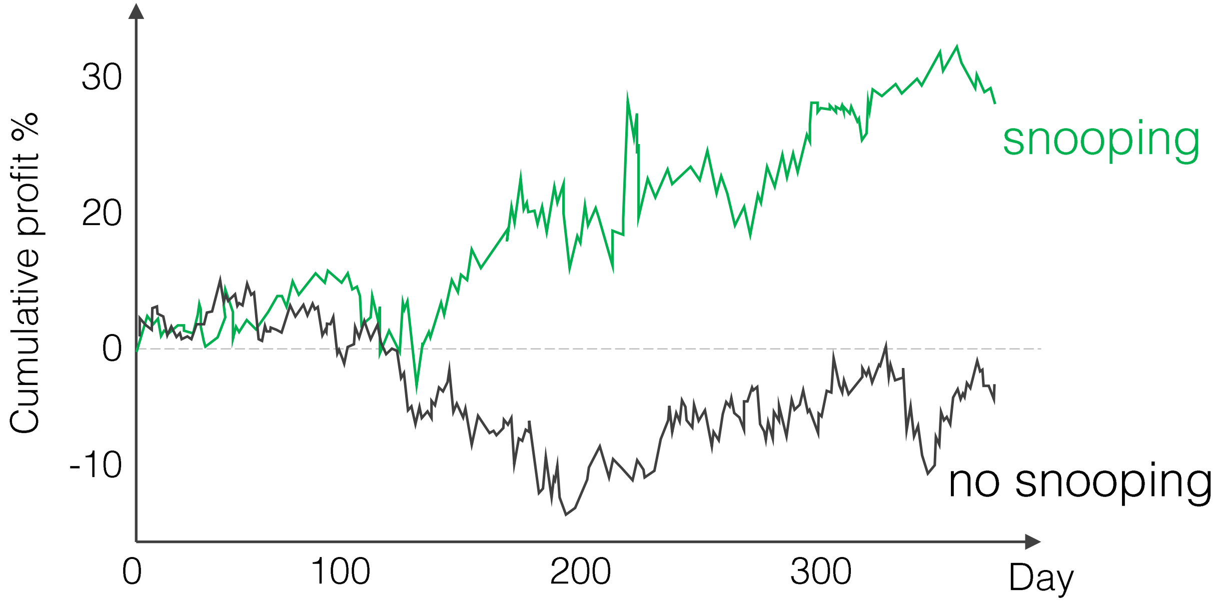
1. Goal: predict the exchange rate for the U.S. Dollar vs British Pound (using 20 past observations)

2. You take your historical data, normalize it, then split it randomly into a training and test set

3. You train on the training data, test on the test data

Results:
Your predictions are correct 56% of the time

Estimate your profits…



Abu-Mostafa, Learning From Data

2

1. Goal: predict the Dow Jones Industrial average

2. You randomly split your data into a training and test dataset

3. Choose a model with lots of flexibility

4. You iterate on the following process dozens of times:
    1. Train your model on the training data
    2. Test your model on the test data
    3. Evaluate performance on the test data

5. Report that you were able to achieve 75% accuracy on your test set!

3

1. Goal: predict long-term performance of a “buy and hold” strategy in stocks

2. You collect 50 years of historical data and include all currently traded companies in the S&P500

3. You randomly split your data into a training and test dataset.

4. You assume you will strictly follow the “buy and hold” strategy

5. You then use apply your model on the current portfolio and predict that you will be rich in retirement!

Abu-Mostafa, Learning From Data

Data snooping

a.k.a. data leakage

If a test data set has affected **any step** in the learning process, its ability to assess the outcome has been **compromised**.

Abu-Mostafa, Learning From Data

Sampling bias

Are the data we're using for machine learning **representative of the population**?

Avoiding data snooping

Don’t touch your test dataset until you’re ready to evaluate your model’s performance

Abu-Mostafa, Learning From Data

Training, Test Split

Learning model parameters

Training	Test
Learn model parameters	Evaluate generalization performance

For small datasets, this reduction in dataset size may be detrimental

Training, Validation, Test Split

Learning parameters AND hyperparameters

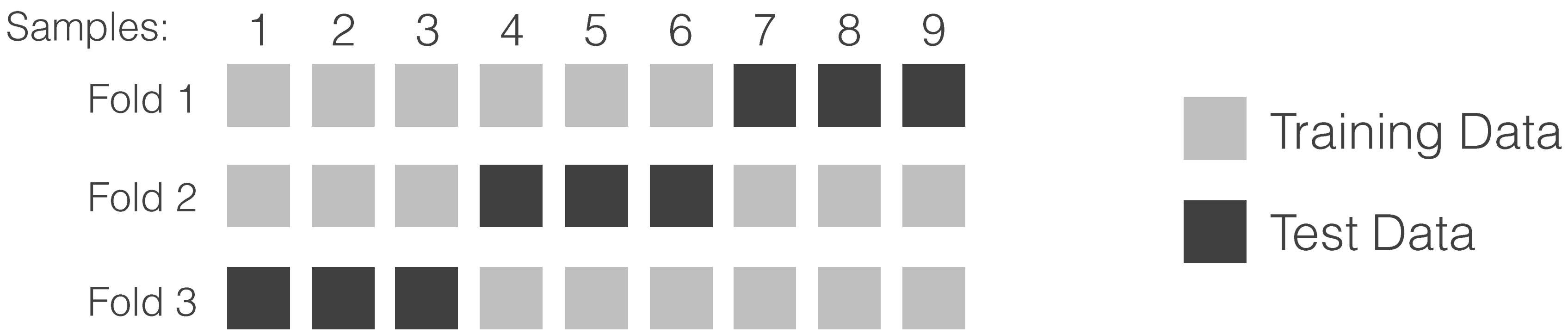
Training	Validation	Test
Learn model parameters	Learn hyperparameters	Evaluate generalization performance

Hyperparameters: parameters of your learning algorithm or parameters of you model that are set before training begins

Simple cross-validation

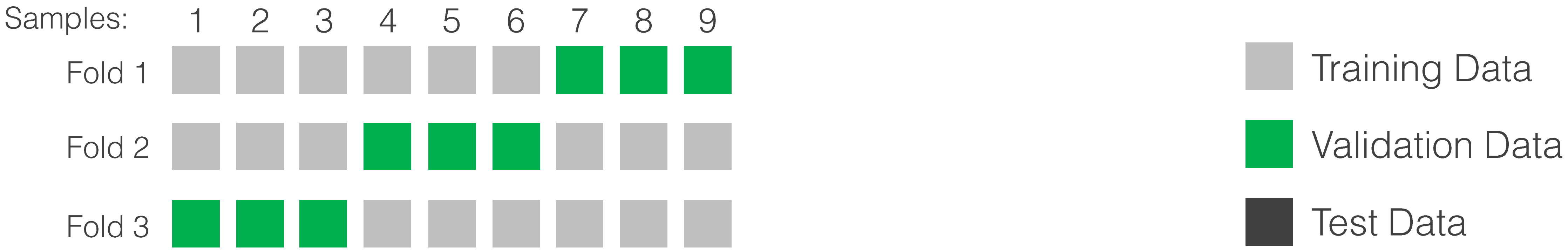
K-fold cross validation   K = 3

1 Performance evaluation: Train your model K times, once for each fold



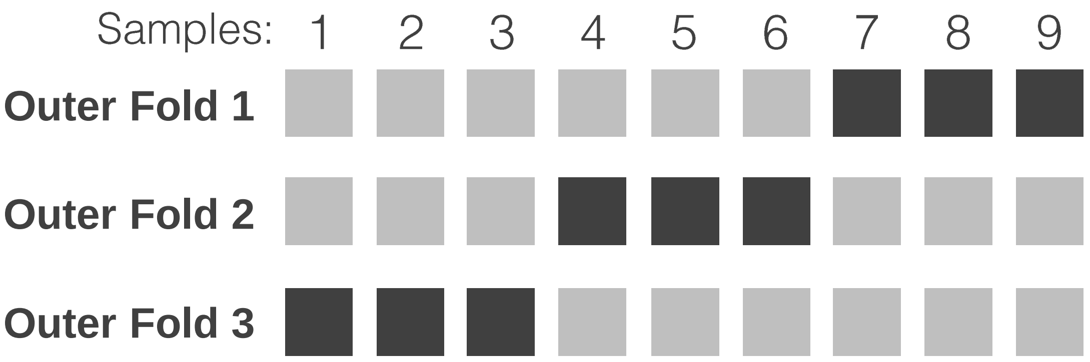
Cross-validation with hyperparameters

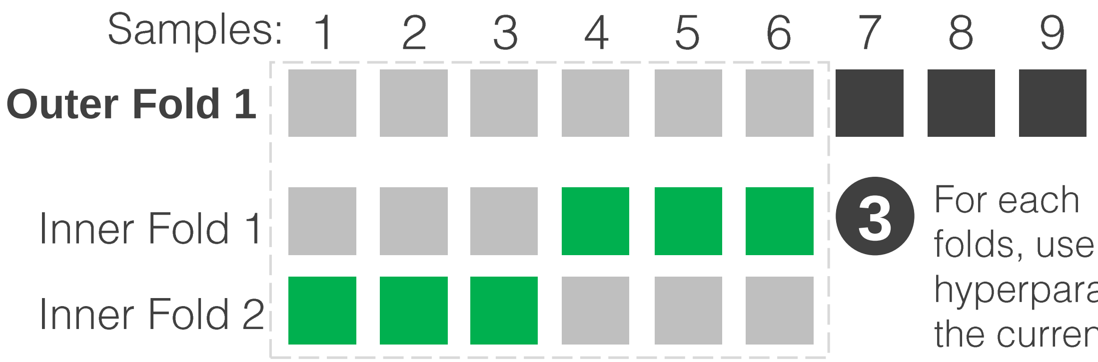
1 Repeatedly fit your model to your K folds. Each iteration try different hyperparameters



2 Using the best-performing hyperparameters from (a), train on all training data and evaluate performance on the test data

Nested cross-validation with hyperparameters





3 For each of the K inner folds, use the hyperparameters from the current iteration to train the model, then evaluate performance on validation data

2 For the inner folds, repeatedly run K-folds, each time with a different set of hyperparameters

1 For each outer fold, train your model with the best-performing hyperparameters from the inner folds

4 Repeat steps (2) and (3) for the remaining outer folds

Training Data

Validation Data

Test Data

Another diagram for nested Cross-validation

Instead of a static train/validation/test split, another option is nested cross-validation

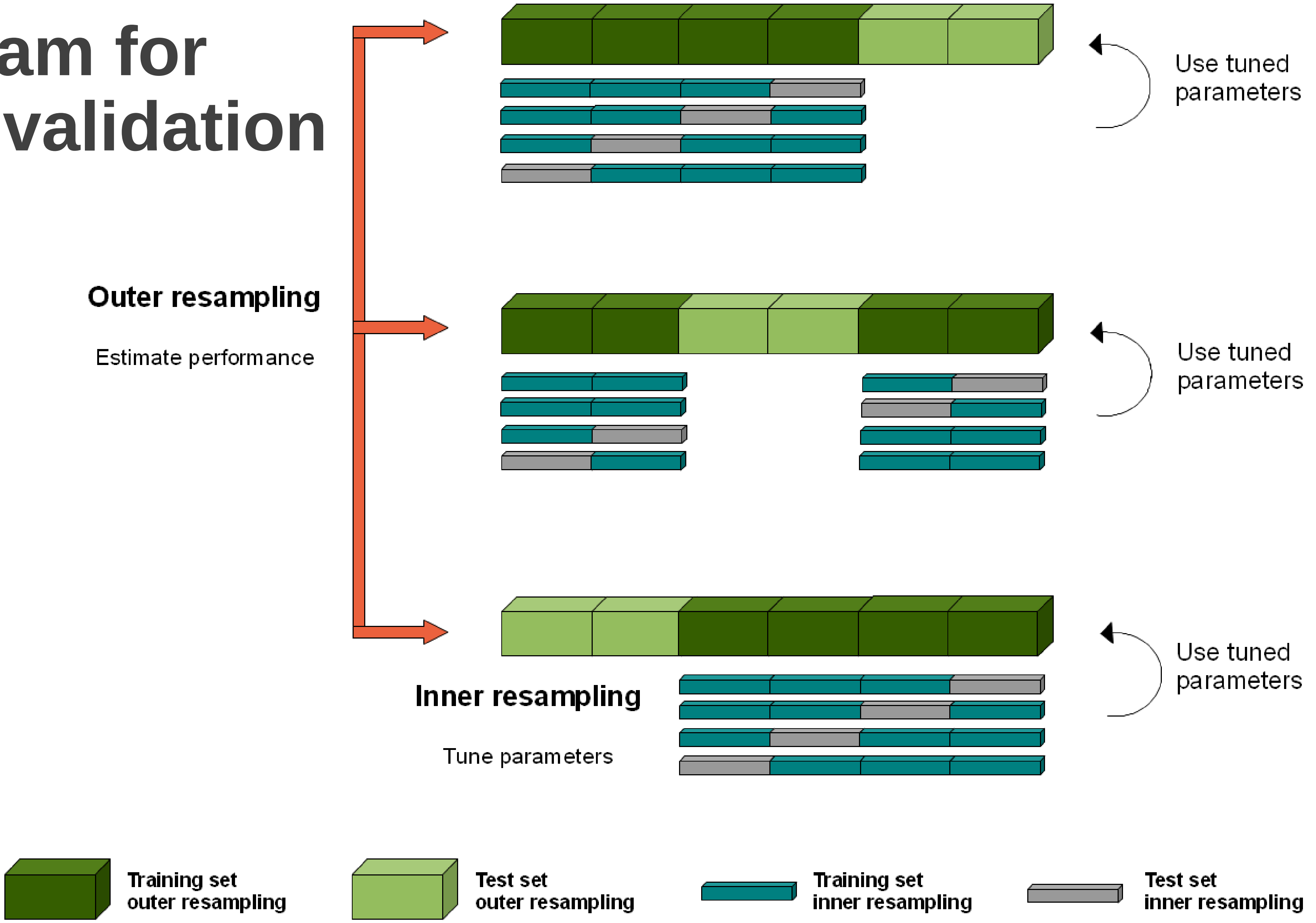
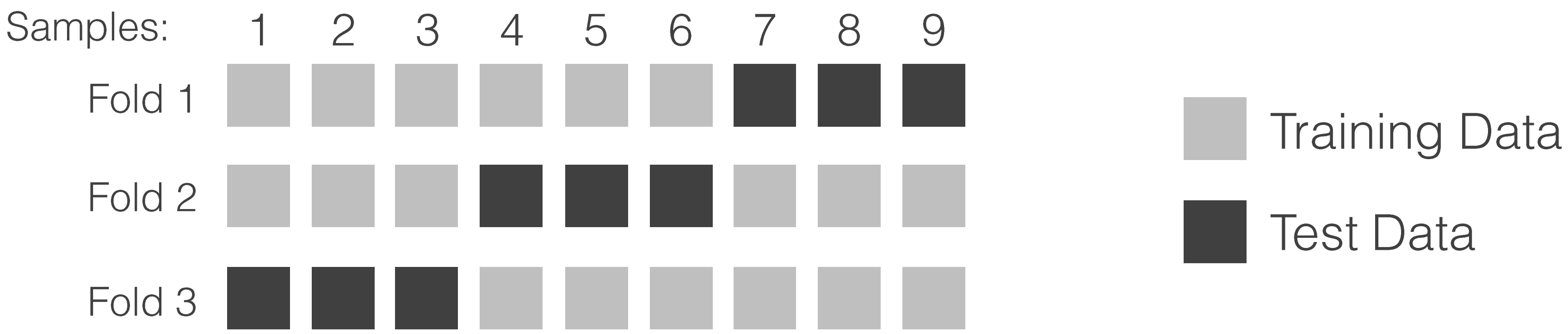


Image Source: https://stats.stackexchange.com/questions/292179/whats-the-meaning-of-nested-resampling

After performance has been validated, train on all the data you have before you apply the model in practice

1 **Performance evaluation**: Train your model K times, once for each fold



2 **Model application**: Once you've evaluated model performance and are ready apply the model then retrain the model on ALL of your data to prepare it for unseen data

(this is not a model evaluation step, but only when you're ready to apply in practice)

But how do I get ROC's out of this?

Each of the K folds will produce a set of confidence scores for the test / validation data of that fold.

Fold 1

y_i	confidence
1	0.98
0	0.87
1	0.43
0	0.02

Fold 2

y_i	confidence
1	0.99
1	0.65
0	0.22
0	0.14

Fold 3

y_i	confidence
1	0.58
0	0.87
0	0.33
0	0.82

1 Merge the outputs from the K folds into a single set of confidence scores for making one ROC curve

Note: The confidence scores need to be on the same scale for this merging method to work properly

y_i	confidence
1	0.98
0	0.87
1	0.43
0	0.02
1	0.99
1	0.65
0	0.22
0	0.14
1	0.58
0	0.87
0	0.33
0	0.82

2 Average the individual ROC curves from each fold

(This also enables measures of variation across the folds)

Note: you only have point data for changes in the ROC curve value, to compute the average you must interpolate between the points on the curve and evaluate the average across all the curves

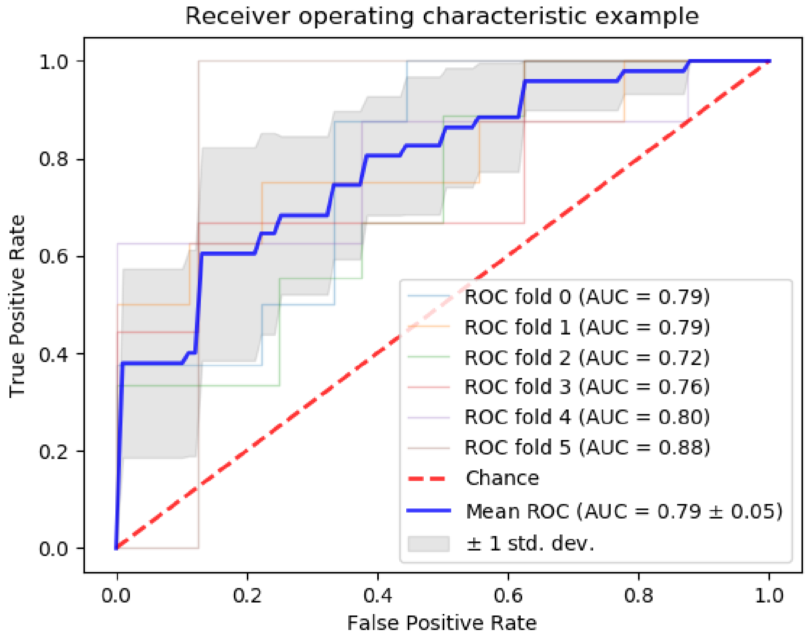


Image from: https://scikit-learn.org/

Bootstrap sampling

Sampling **with replacement**

Often used to estimate standard errors and confidence intervals

Integral part of model ensembles (i.e. bagging in random forests)

Abu-Mostafa, Learning From Data