CDS

Cornell Data Science

# Cross Validation

# Generally: Cross Validation (CV)

Set of **validation techniques** that use the training dataset itself to validate model

- Allows maximum allocation of training data from original dataset
- Efficient due to advances in processing power

Cross validation is used to test the effectiveness of any model or its modified forms.

# Validation Goal

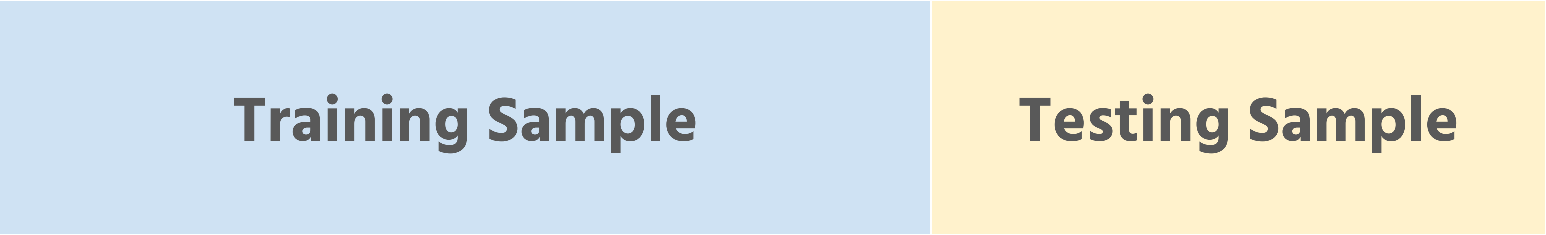
- Estimate Expected Prediction Error
- Best Fit model
- Make sure that the model does not Overfit

Hastie et al. “Elements of Statistical Learning.”

# HoldOut Validation

Dataset

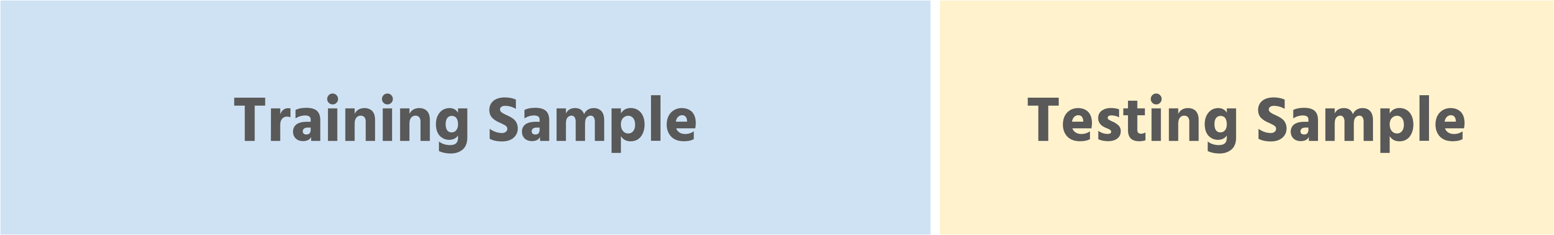
# HoldOut Validation



Training Sample

Testing Sample

# HoldOut Validation



Training Sample

Testing Sample

Advantage: Traditional and Easy
Disadvantage: Varying Error based on how to sample testing

# *K*-fold Validation

Create equally sized *k* partitions, or **folds**, of training data

For each fold:

- Treat the *k-1* other folds as training data.
- Test on the chosen fold.

The average of these errors is the validation error

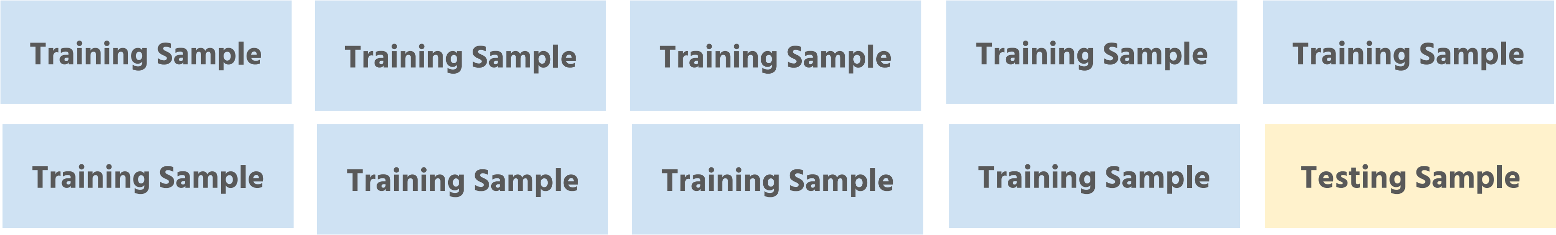
Often used in practice with *k*=5 or *k*=10.

# *K*-fold Validation

**Suppose K = 10,**

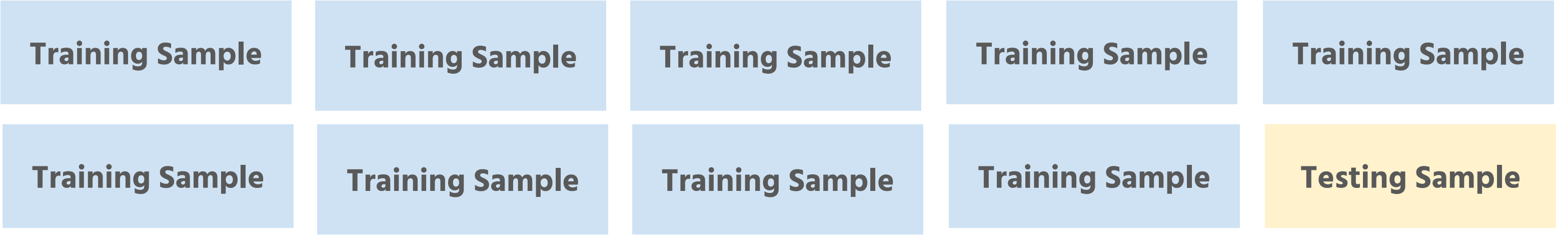
**10-Fold CV**

# *K*-fold Validation



| | | | | |
|---|---|---|---|---|
| Training Sample | Training Sample | Training Sample | Training Sample | Training Sample |
| Training Sample | Training Sample | Training Sample | Training Sample | Testing Sample |

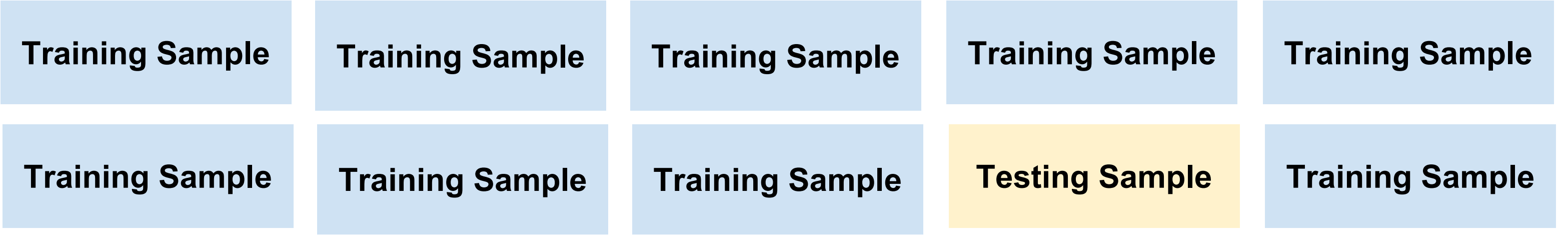
# *K*-fold Validation



| | | | | |
|---|---|---|---|---|
| Training Sample | Training Sample | Training Sample | Training Sample | Training Sample |
| Training Sample | Training Sample | Training Sample | Training Sample | Testing Sample |

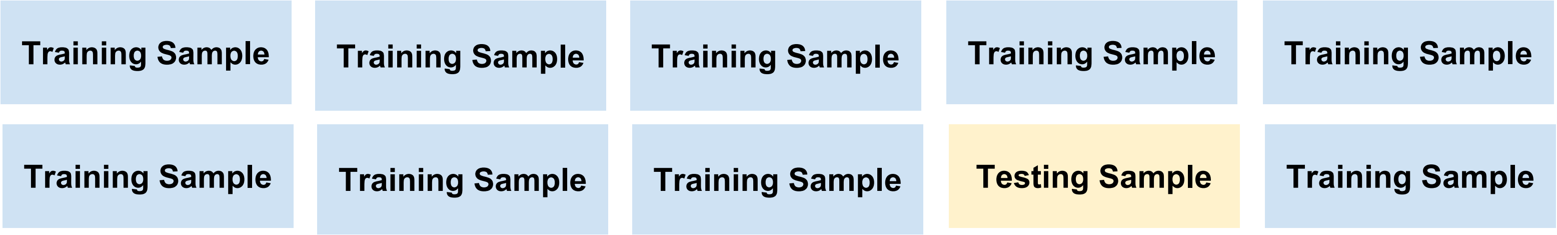
**Calculate RMSE = rmse1**

# *K*-fold Validation



| | | | | |
|---|---|---|---|---|
| Training Sample | Training Sample | Training Sample | Training Sample | Training Sample |
| Training Sample | Training Sample | Training Sample | Testing Sample | Training Sample |

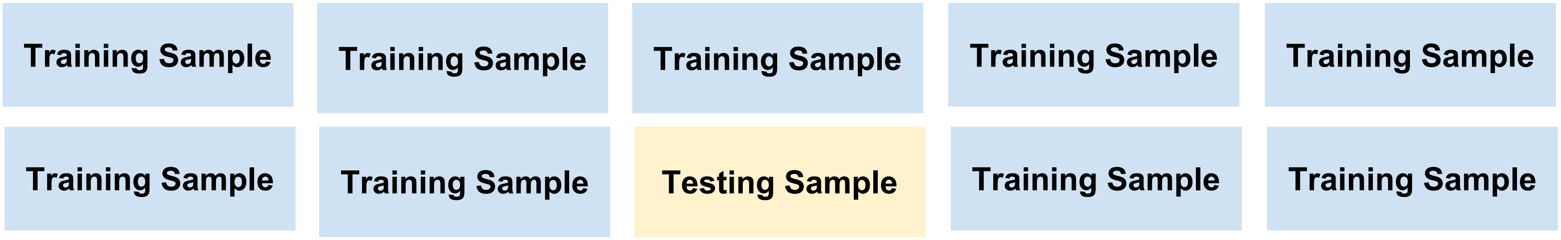
# *K*-fold Validation



| | | | | |
|---|---|---|---|---|
| Training Sample | Training Sample | Training Sample | Training Sample | Training Sample |
| Training Sample | Training Sample | Training Sample | Testing Sample | Training Sample |

**Calculate RMSE = rmse2**

# *K*-fold Validation



| | | | | |
|---|---|---|---|---|
| Training Sample | Training Sample | Training Sample | Training Sample | Training Sample |
| Training Sample | Training Sample | Testing Sample | Training Sample | Training Sample |

# *K*-fold Validation

| | | | | |
|---|---|---|---|---|
| Training Sample | Training Sample | Training Sample | Training Sample | Training Sample |
| Training Sample | Training Sample | Testing Sample | Training Sample | Training Sample |

Calculate RMSE = rmse3

# *K*-fold Validation

**And so on**

# *K*-fold Validation

| | | | | |
|---|---|---|---|---|
| Testing Sample | Training Sample | Training Sample | Training Sample | Training Sample |
| Training Sample | Training Sample | Training Sample | Training Sample | Training Sample |

**Calculate RMSE = rmse10**

# *K*-fold Validation

| | | | | |
|---|---|---|---|---|
| Testing Sample | Training Sample | Training Sample | Training Sample | Training Sample |
| Training Sample | Training Sample | Training Sample | Training Sample | Training Sample |

RMSE = Avg(rmse1…10)

# *K*-fold Validation

Less matters how we divide up

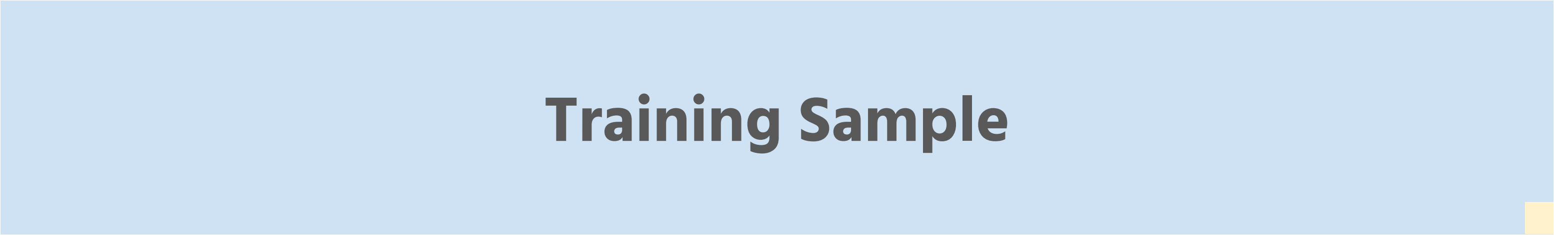
Selection bias not present

# Leave-One-Out Method

# Leave-One-Out Method

Dataset

# Leave-One-Out Method

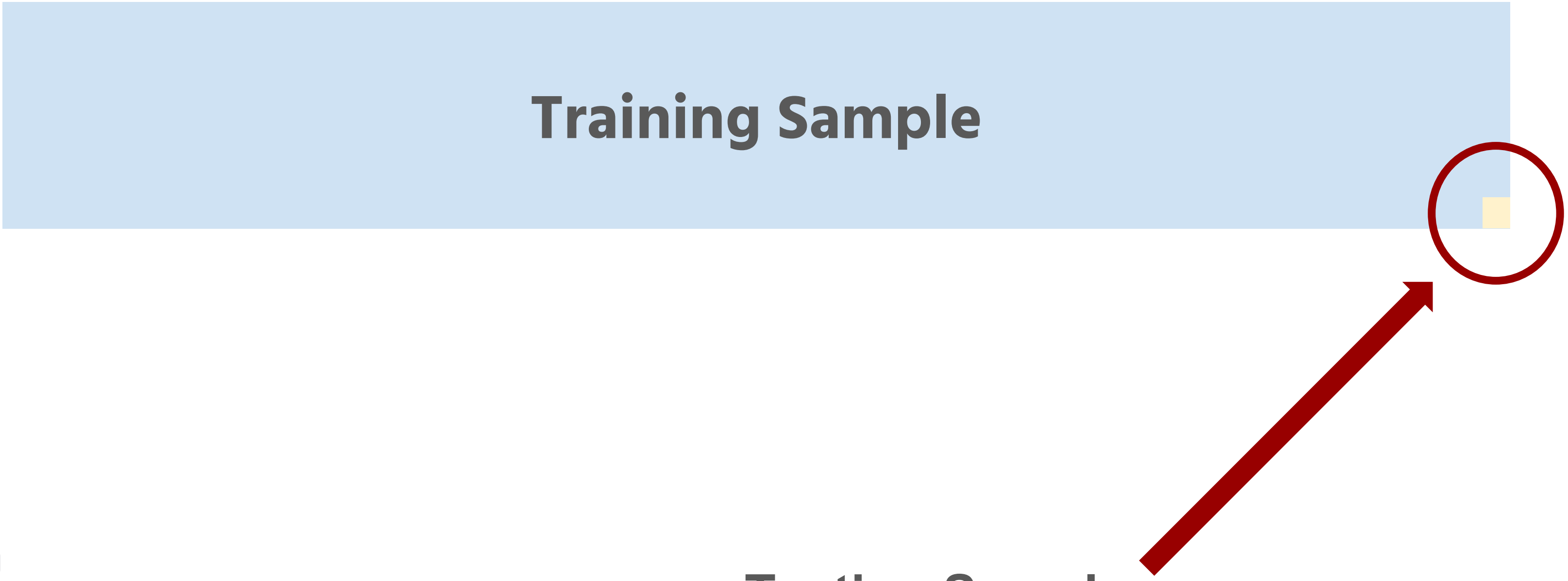


Training Sample

## Leave-One-Out Method

What just happened?

# Leave-One-Out Method



Training Sample

Testing Sample

# Leave-P-Out Validation

For each data point:

- Leave out p data points and train learner on the rest of the data.
- Compute the test error for the p data points.

Define average of these ${}_{n}C_{p}$ error values as validation error

Source

# Leave-P-Out Validation

A really exhaustive and thorough way to validate

High Computation Time

**Question:**

How are *k*-fold and leave-p-out different?

# Subset Selection

- **Best subset selection:** Test all $2^p$ subset selections for best one
- **Forward subset selection**
  - Iterate over k = 0 … (p-1) predictors
  - At each stage, select the best model with (p-k) predictors
  - Find best model out of the p-1 selected candidates with CV
- **Backward selection** - Reverse of forward subset selection
  - Start from p predictors and work down

In practice, best subset selection method is rarely used, why?

# Coming Up

**Your problem set:** Project E

**Next week:** Thanksgiving?!

Aw yeah.

Cornell Data Science