

# Cross Validation

---

# OLS Solution

- Ordinary Least Squares (OLS)

Regression:

$$\min_{\beta_0 \dots \beta_k} \sum_i [y_i - (\beta_0 + \beta_1 x_{i,1} + \dots + \beta_k x_{i,k})]^2$$

where  $x_{i,j}$  is the value in  $i$ th row and  $j$ th column of  $X$ . i.e. the  $i$ th value in the  $j$ th independent variable.

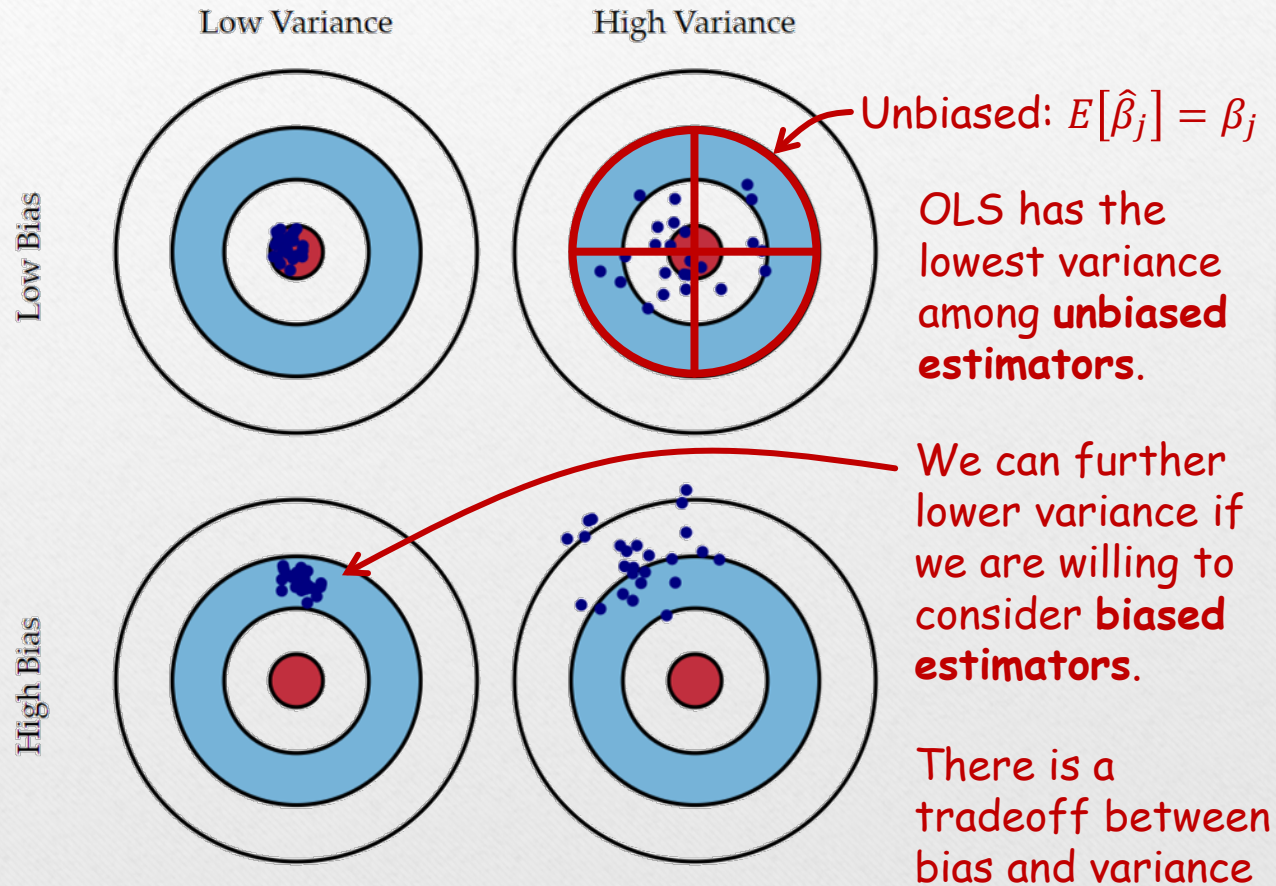
- Solution to this problem can be represented by the following matrix equation:

$$\vec{\beta} = (X^T X)^{-1} X^T \vec{y}$$



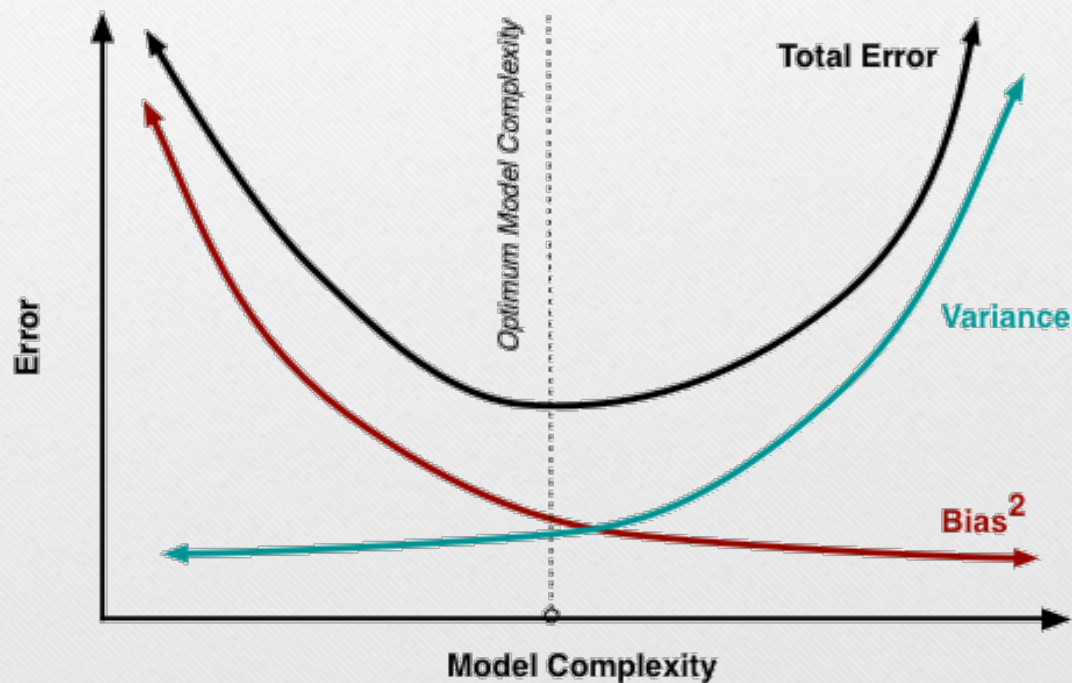


# Variance vs Bias



# Variance-Bias Tradeoff

- The tradeoff between a model's ability to minimize variance and to minimize bias



Source: <http://scott.fortmann-roe.com/docs/BiasVariance.html>



# Regularization

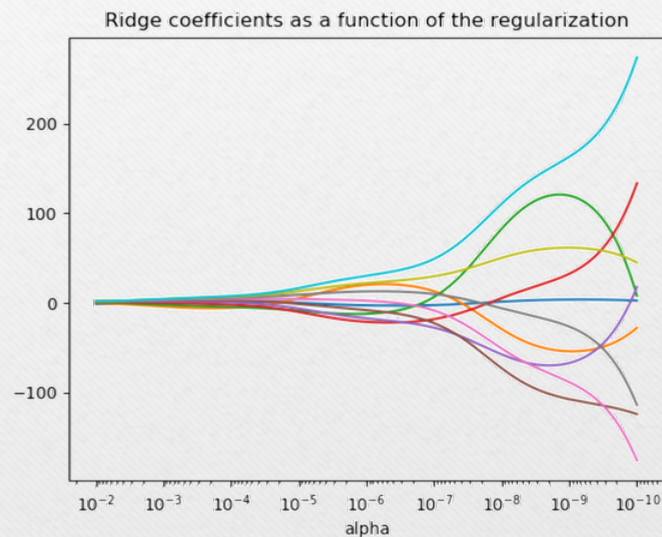
- **Regularization** penalizes large coefficients
- This reduce overfitting by making regression coefficients over different samples more similar to each other
  - In the most extreme case, all coefficients will be zero regardless of sample
- Regularized regression are biased, but they have smaller variance than OLS
- The two most regularized regressions are **ridge** and **lasso**

# Regularization

- Ridge Regression, also called **Least Square with L2-regularization**, have the following objective:

$$\min_{\beta} \left\{ \sum_i (y_i - \vec{\beta} \cdot \vec{x}_i)^2 + \alpha \sum_k \beta_k^2 \right\}$$

- Ridge regression pushes **all** coefficients **towards zero**
- Stronger regularization (higher  $\alpha$ ) leads to smaller coefficients





# Regularization

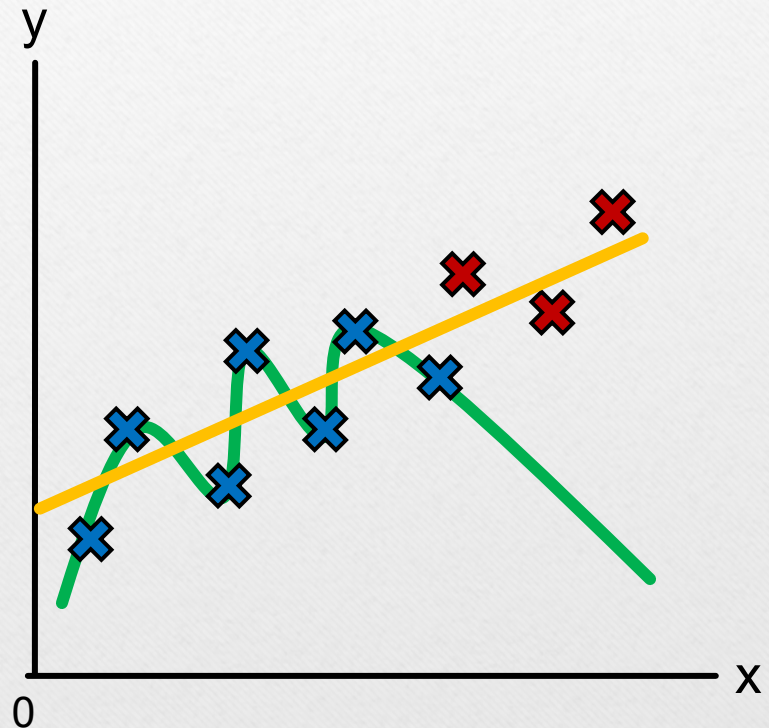
- Lasso Regression, also called **Least Square with L1-regularization**, have the following objective:

$$\min_{\beta} \left\{ \sum_i (y_i - \vec{\beta} \cdot \vec{x}_i)^2 + \alpha \sum_k |\beta_k| \right\}$$

- Lasso regression makes **some** coefficients **exactly zero**
- Stronger regularization (higher  $\alpha$ ) leads to more coefficients becoming zero
- **Lasso can help you select variables**

# Out-of-Sample Test

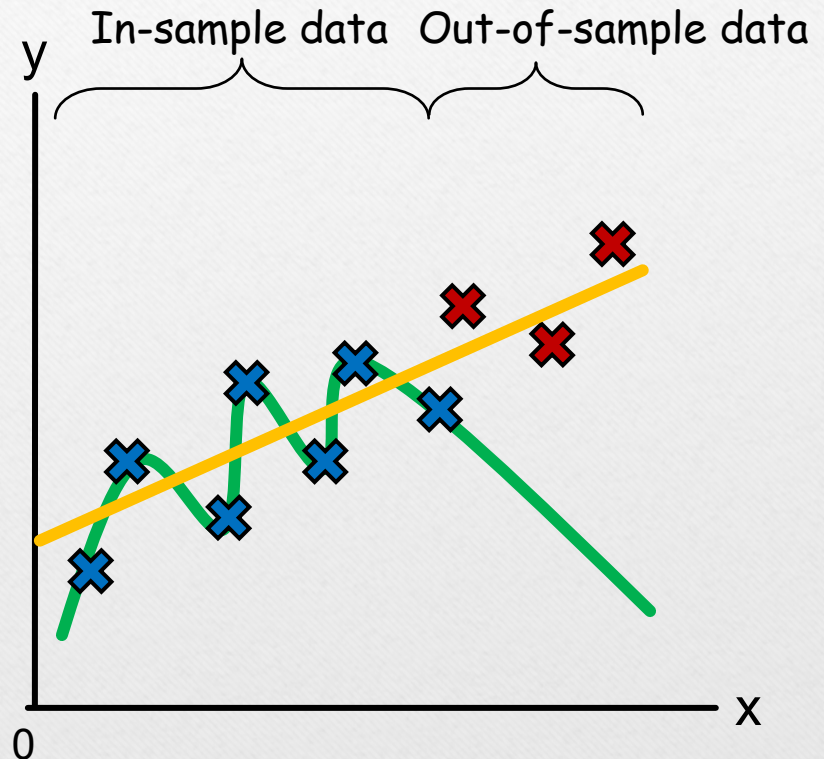
- We want accurate prediction
- Given enough complexity, a model will always do well with data it has seen
- We want to know if the model does well with data it has **not** yet seen





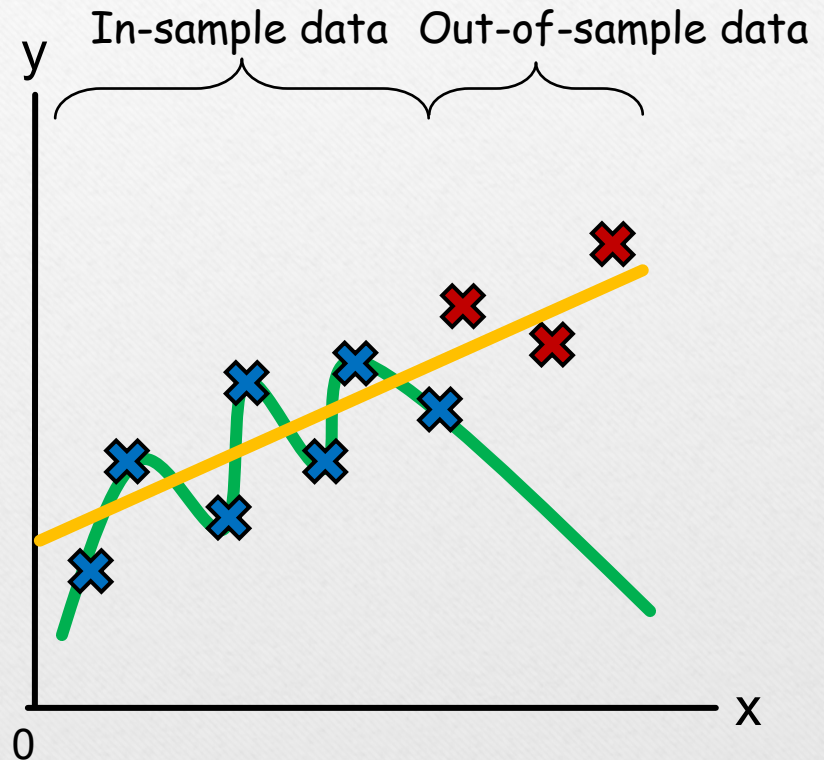
# Out-of-Sample Test

- We will intentionally keep part of the data from the model training process
- This reserved data is not seen by the model during training, allowing us to conduct an **out-of-sample test**
- We pick the model (or model parameters) that has the highest out-of-sample prediction accuracy



# How to Find the Best $\alpha$ ?

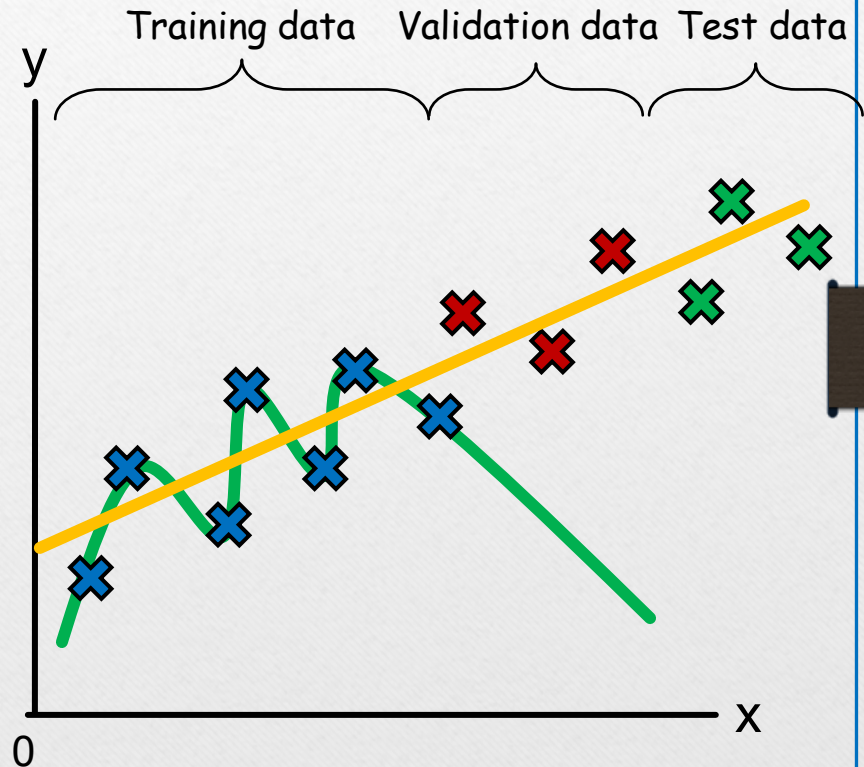
- Idea: we pick the model (or model **hyperparameters** such as  $\alpha$ ) that has the highest out-of-sample prediction accuracy
- Problem: If we pick  $\alpha$  this way, the model has seen the reserved data, so it is no longer out of sample





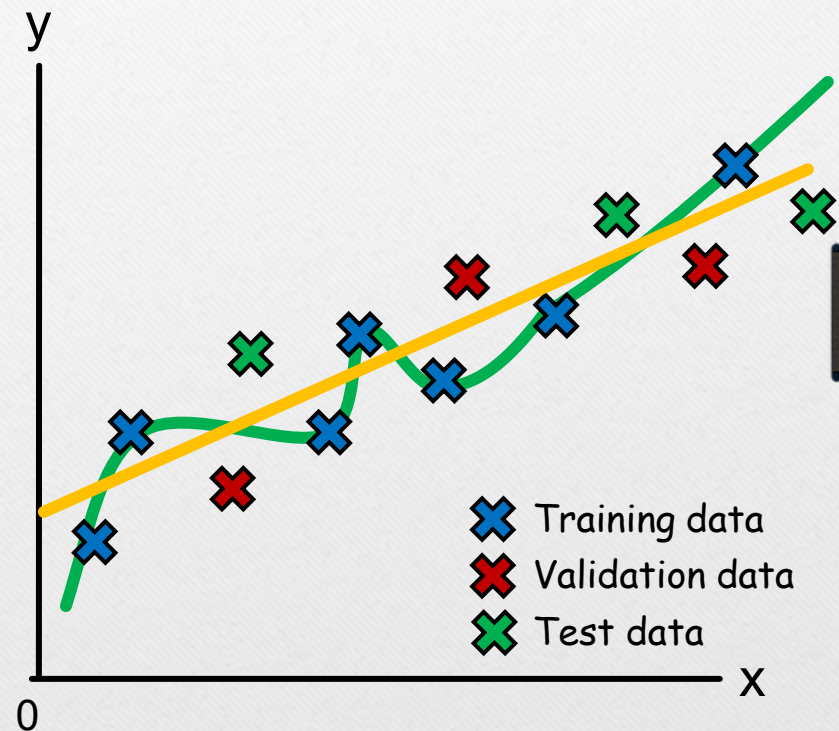
# How to Find the Best Model

- Solution: split the data into three parts:
  1. **Training set** for training the model
  2. **Validation set** for choosing models and hyperparameters
  3. **Test set** for reporting out-of-sample performance



# How to Find the Best Model

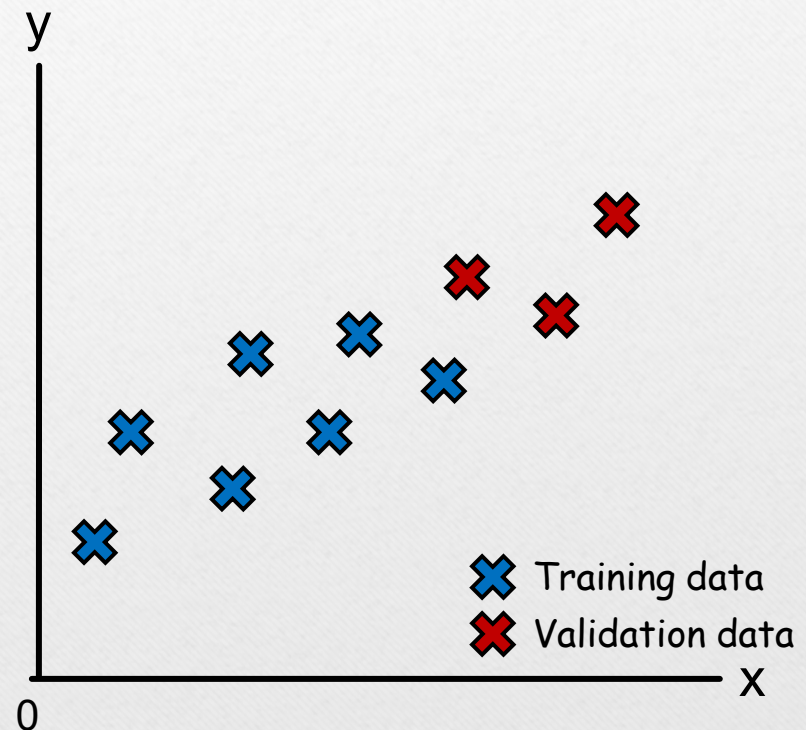
- To ensure that each set of data is representative, we generally want to split the data randomly rather than sequentially
- The exception is time series data. We must split such data sequentially to avoid hindsight bias





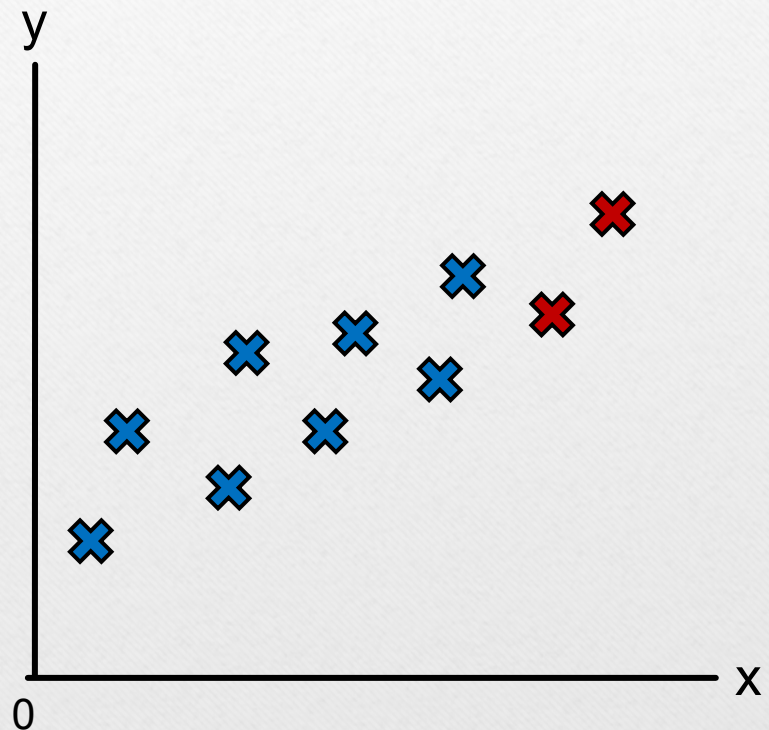
# K-Fold Cross Validation

- We might question the representativeness of the validation set, particularly when the sample size is small
- At the same time, we might be unwilling to increase its size since that makes the training set smaller
- What to do in this case?



# K-Fold Cross Validation

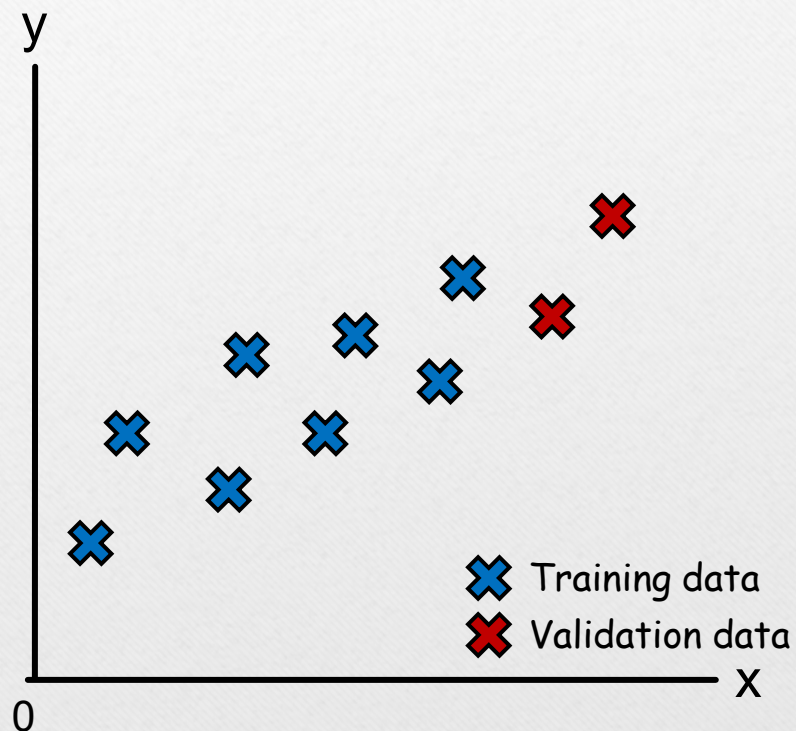
- We can handle this problem by using ***k*-fold cross validation**
- We first divide the data we intend to use for training and validation into  $k$  equal-sized folds. Five is a common choice
- We repeat the training process for  $k$  times. Each time we use one fold for validation and the rest for training





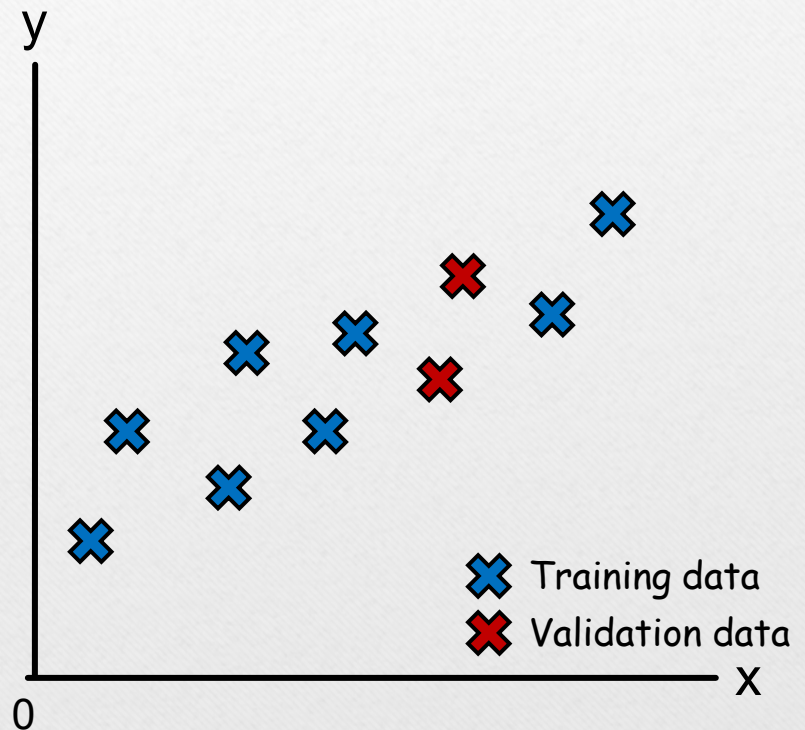
# *K*-Fold Cross Validation

- Iteration 1



# *K*-Fold Cross Validation

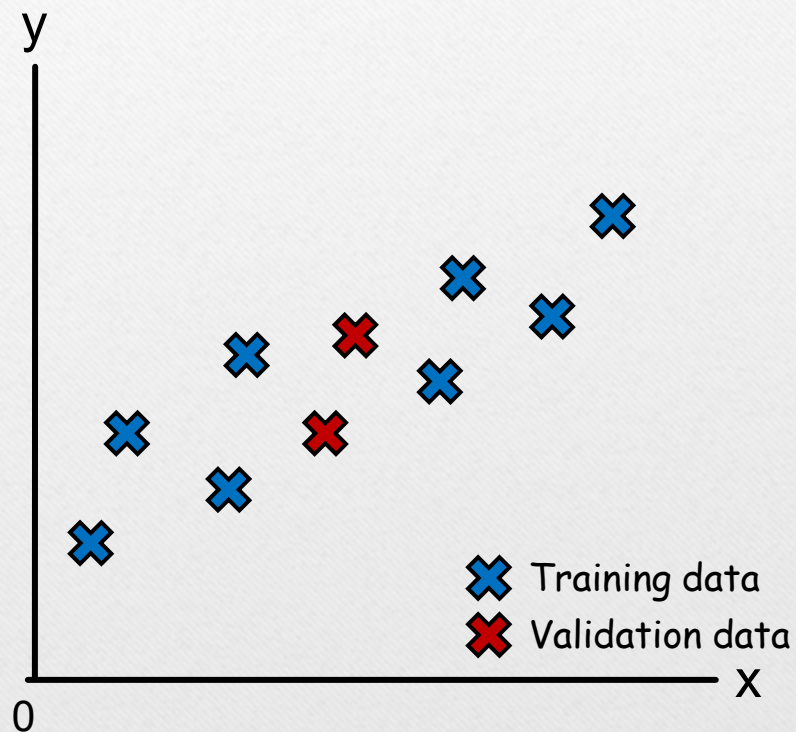
- Iteration 2





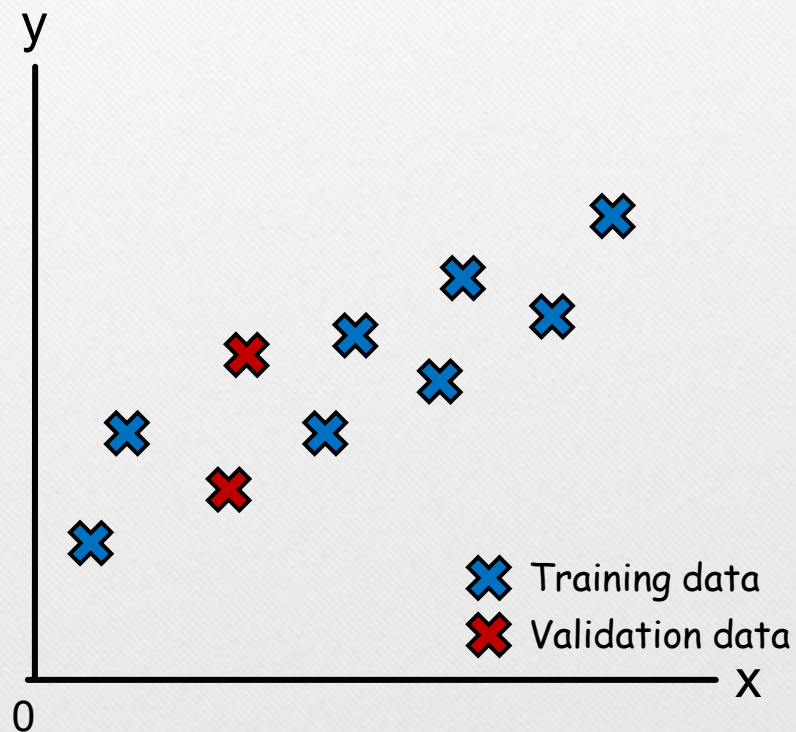
# *K*-Fold Cross Validation

- Iteration 3



# *K*-Fold Cross Validation

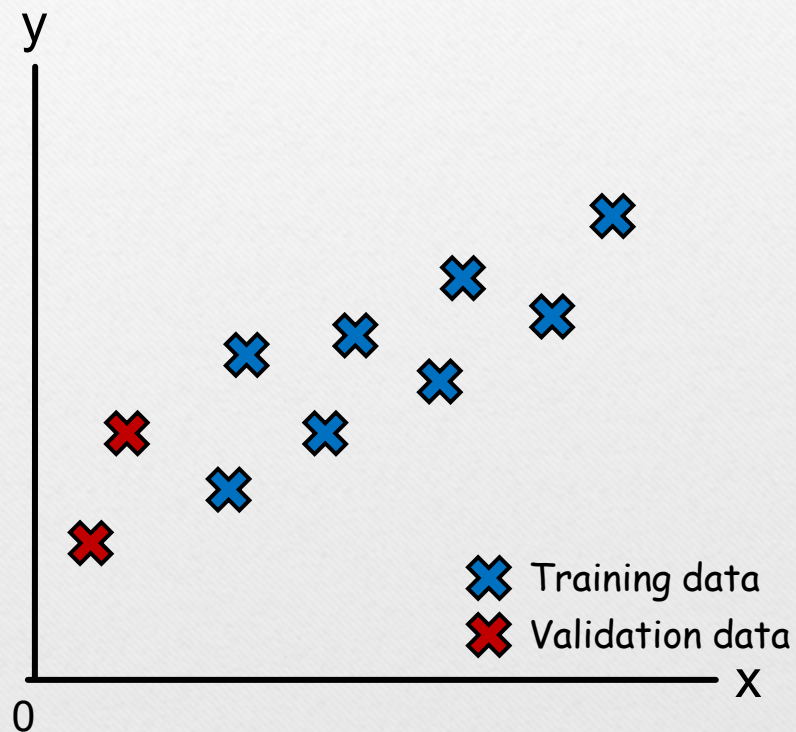
- Iteration 4





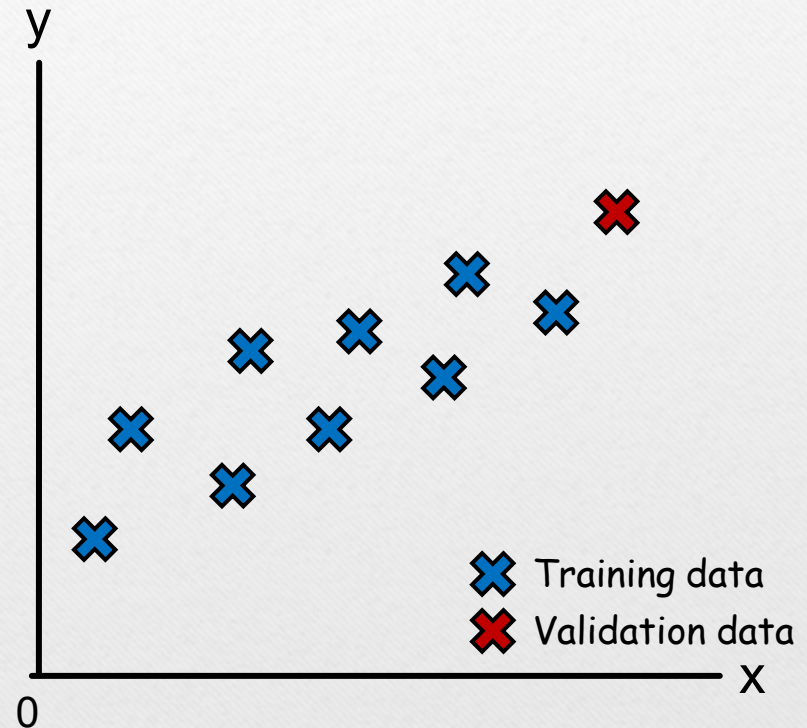
# *K*-Fold Cross Validation

- Iteration 5



# Leave-One-Out Validation

- If each fold only has one sample, the process is called leave-one-out validation (LOOV)





# K-Fold Cross Validation

- Performance measures are averaged across folds
- K-fold cross validation trades training time for representativeness of validation data
- Might not be feasible when model is time consuming to train

