Error Analysis

Bias and Variance

Example: Linear regression (housing prices)

$\theta_0 + \theta_1 x$

Fitting a linear function

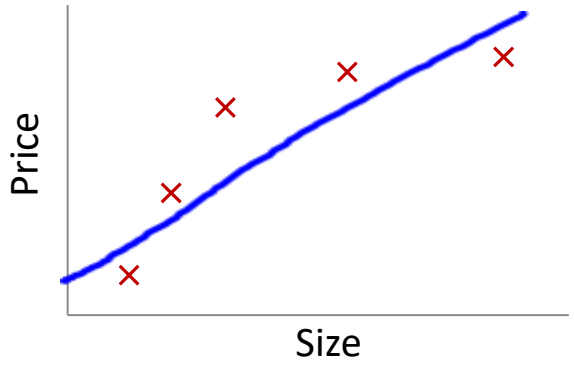
$\theta_0 + \theta_1 x + \theta_2 x^2$

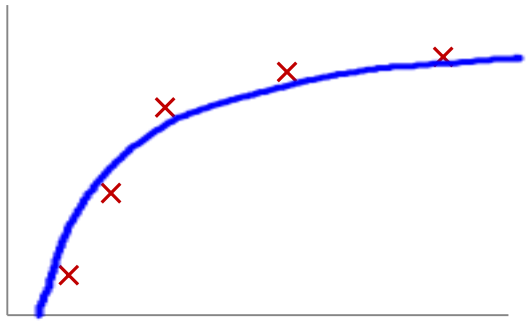
Fitting a quadratic function

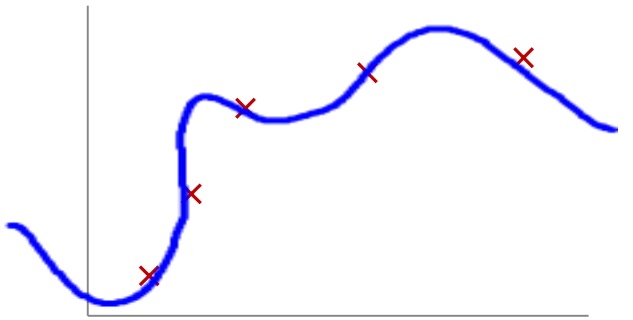
$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

Fitting a higher order function

Bias vs. variance in linear regression







$d = 1$

$$\theta_0 + \theta_1 x$$

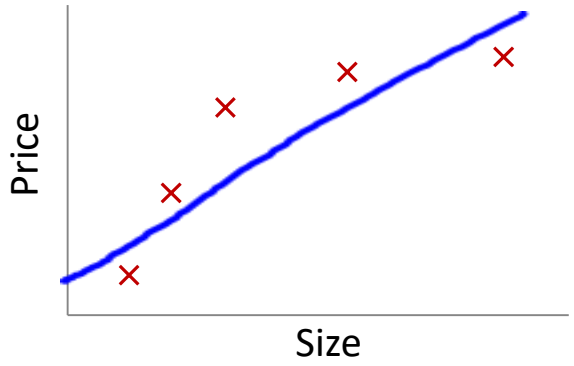
$d = 2$

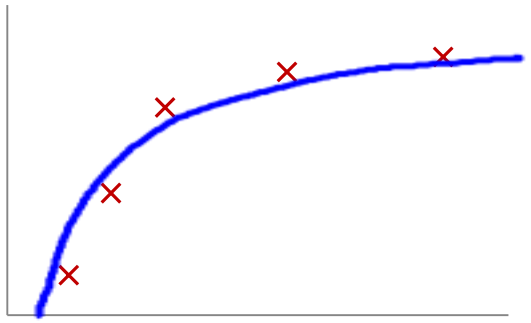
$$\theta_0 + \theta_1 x + \theta_2 x^2$$

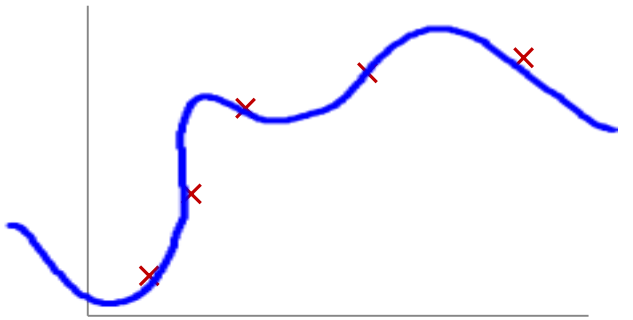
$d = 4$

$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Bias vs. variance in linear regression







High bias
(underfitting)

$d = 1$

$$\theta_0 + \theta_1 x$$

"Just right"

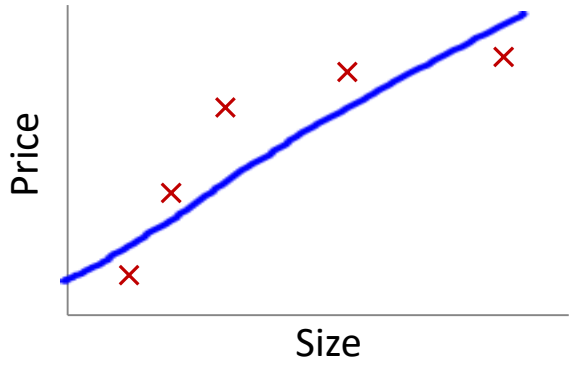
$d = 2$

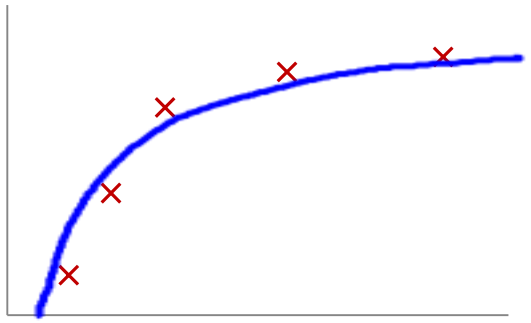
$$\theta_0 + \theta_1 x + \theta_2 x^2$$

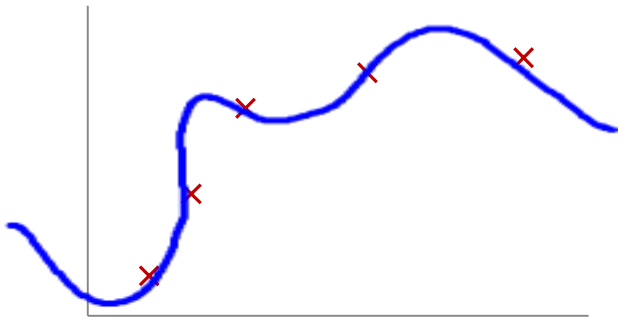
$d = 4$

$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Bias vs. variance in linear regression







High bias
(underfitting)

$d = 1$

$\theta_0 + \theta_1 x$

"Just right"

$d = 2$

$\theta_0 + \theta_1 x + \theta_2 x^2$

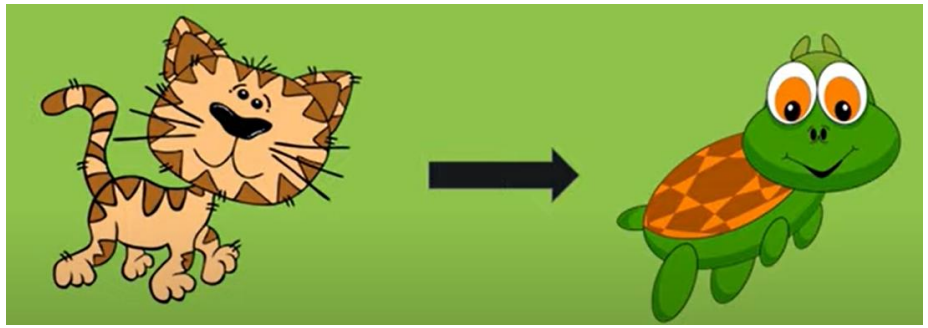
High variance
(overfitting)

$d = 4$

$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

What is Bias?

- Bias is the difference between the average prediction of our model and the correct value which we are trying to predict. Model with high bias pays very little attention to the training data and oversimplifies the model. It always leads to high error on training and test data. The model is underfitted.

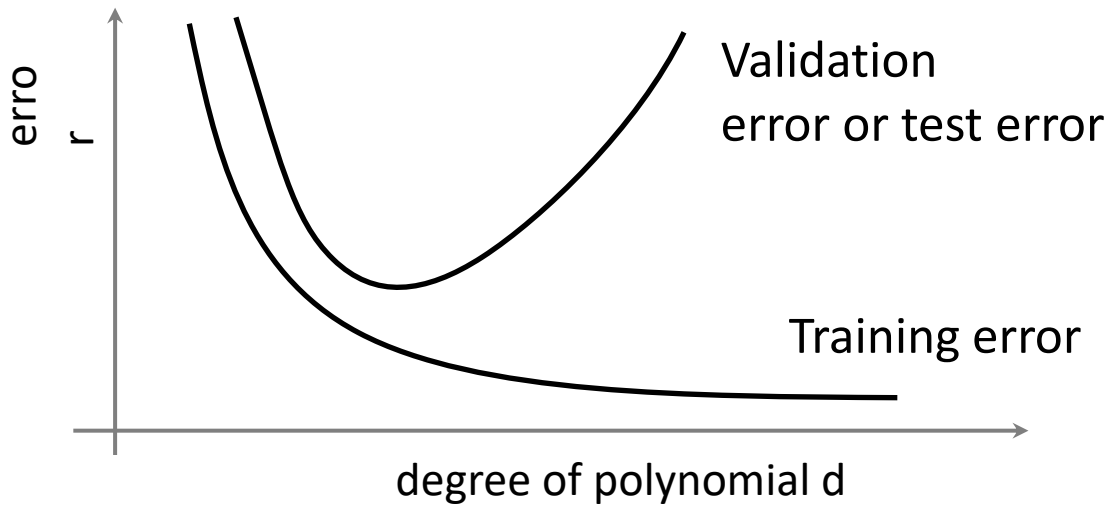


Varience

- Variance is the variability of model prediction for a given data point or a value which tells us spread of our data.
- How scattered the predicted values are from actual values
- Variance signifies the model is trained with a lot of noise and irrelevant data
- Model with high variance pays a lot of attention to training data and does not **generalize** on the data which it hasn’t seen before. As a result, such models perform very well on training data but has high error rates on test data.

Analysing bias vs. variance

- Suppose your model is not performing as well as expected. Is it a bias problem or a variance problem?



Bias (underfit):
Both training error and validation / test error are high

Variance (overfit):
Low training error
High validation / test error

Bias vs. Variance

- Bias and variance both contribute to the error of classifier
- Variance is error due to randomness in how the training data was selected (variance of an estimate refers to how much the estimate will vary from sample to sample)
- Bias is error due to something systematic, not random

Will more training data help?

- A learnt model is not performing as well as expected. Will having more training data help?

- Note that there can be substantial cost for getting more training data.

Will more training data help?

- A learnt model is not performing as well as expected. Will having more training data help?

- Note that there can be substantial cost for getting more training data.

- If model is suffering from high bias, getting more training data will not (by itself) help much.
- If model is suffering from high variance, getting more training data is likely to help

Overfitting

If we have too many features, the learned hypothesis may fit the training set very well

but fail to generalize to new examples.

Overfitting and Underfitting

- Underfitting: model is too simple to represent all the relevant class characteristics

- Overfitting: model is too complex and fits irrelevant characteristics (noise) in the data

Overfitting

- Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. This means that the noise or random fluctuations in the training data is picked up and learned as concepts by the model. The problem is that these concepts do not apply to new data and negatively impact the models ability to generalize

Underfitting

- Underfitting refers to a model that can neither model the training data nor generalize to new data.
- An underfit machine learning model is not a suitable model and will be obvious as it will have poor performance on the training data.

Trade-0ff

- In machine learning, there is always a trade-off between

– complex hypotheses that fit the training data well

– simpler hypotheses that may generalise better.

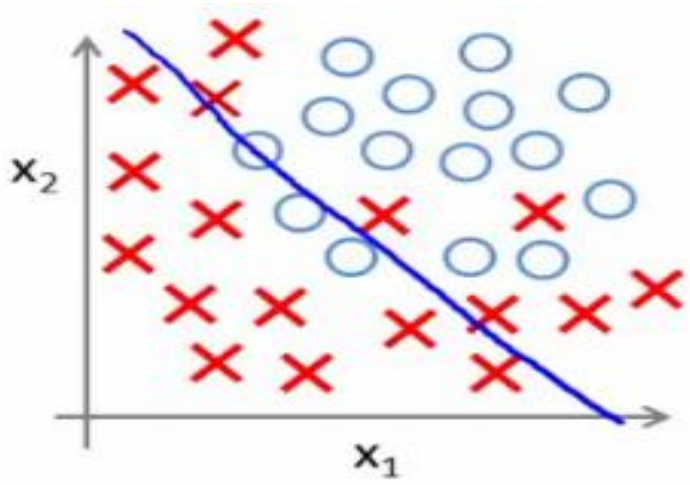
As the amount of training data increases, the generalization error decreases.

A Good Fit in Machine Learning

- Ideally, you want to select a model at the sweet spot between underfitting and overfitting.

Bias vs. variance in logistic regression

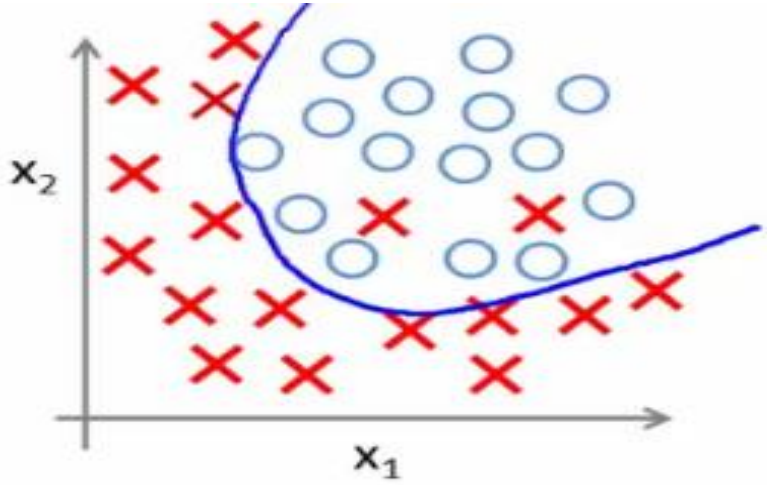
Example: Logistic regression



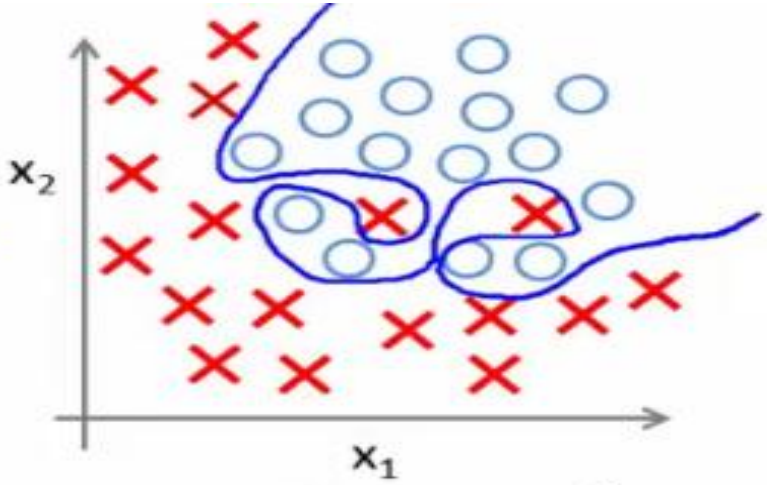
$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$

(g = sigmoid function)

UNDERFITTING
(high bias)



$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$



$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \ldots)$

OVERFITTING
(high variance)

Ways to handle noise

- Validation
  - Check performance on data other than training data, and tune model accordingly
- Regularization
  - Constraint the model so that the noise cannot be learnt too well

Validation

Using Validation Set

- Divide training data into two parts:
  - Training set:
    - use for model building
  - Validation set:
    - use for estimating generalization error
    - Note: validation set is not the same as test set

- Drawback:
  - Less data available for training

Validation

- Divide given data into train set and test set
  - E.g., 80% train and 20% test
  - Better to select randomly
- Learn parameters using training set
- Check performance (validate the model) on test set, using measures such as accuracy, misclassification rate, etc.
- Trade-off: more data for training vs. validation

An example: model selection

- Which order polynomial will best fit a given data? Polynomials available: h_1, h_2, …, h_{10}
- As if an extra parameter - degree of the polynomial -is to be learned
- Approach
  - Divide into train and test set
  - Train each hypothesis on train set, measure error on test set
  - Select the hypothesis with minimum test set error

An example: model selection

- Problem with the previous approach
  - The test set error we computed is not a true estimate of generalization error
  - Since our extra parameter (order of polynomial) is fit to the test set

An example: model selection

- Approach 2
  - Divide data into train set (60%), validation set (20%) and test set (20%)
  - Select that hypothesis which gives lowest error on validation set
  - Use test set to estimate generalization error

- Note: Test set not at all seen during training

Popular methods of evaluating a classifier

- Holdout method
  - Split data into train and test set (usually 2/3 for train and 1/3 for test). Learn model using train set and measure performance over test set
  - Usually used when there is sufficiently large data, since both train and test data will be a part

Popular methods of evaluating a classifier

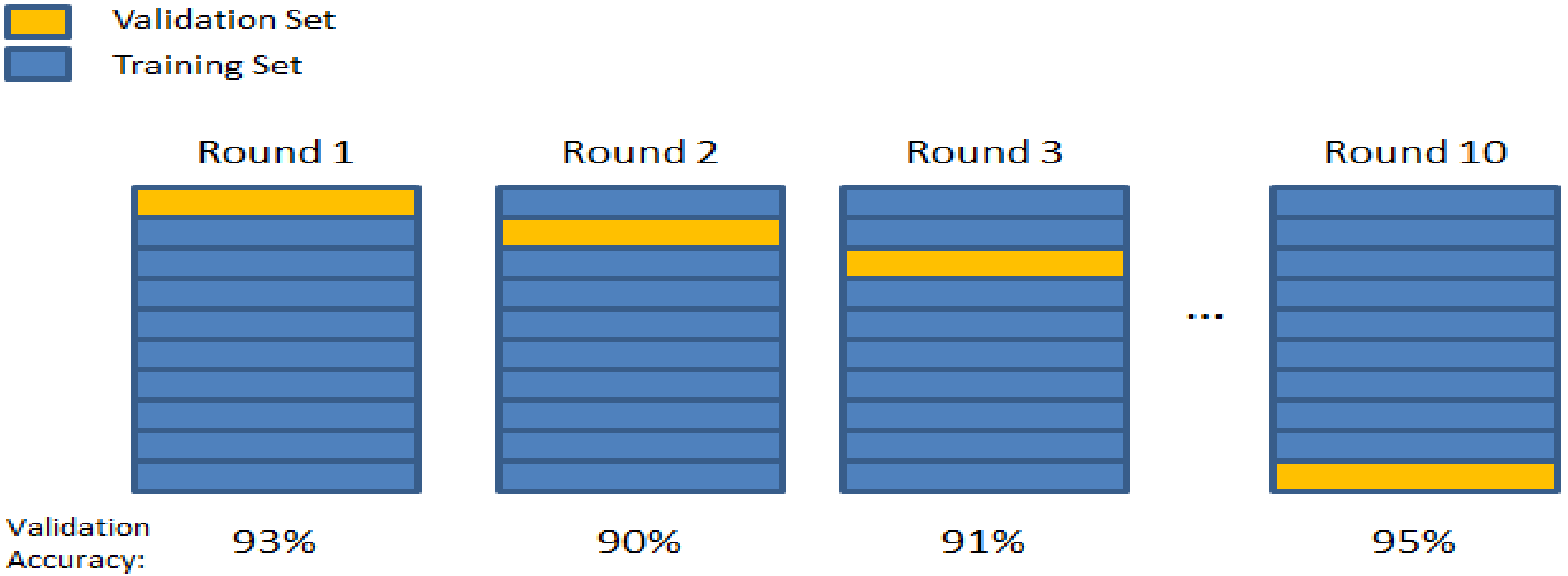
- Repeated Holdout method
  - Repeat the Holdout method multiple times with different subsets used for train/test
  - In each iteration, a certain portion of data is randomly selected for training, rest for testing
  - The error rates on the different iterations are averaged to yield an overall error rate
  - More reliable than simple Holdout

Popular methods of evaluating a classifier

- k-fold cross-validation
  - *First step*: data is split into *k* subsets of equal size;
  - *Second step*: each subset in turn is used for testing and the remainder for training
  - Performance measures averaged over all folds

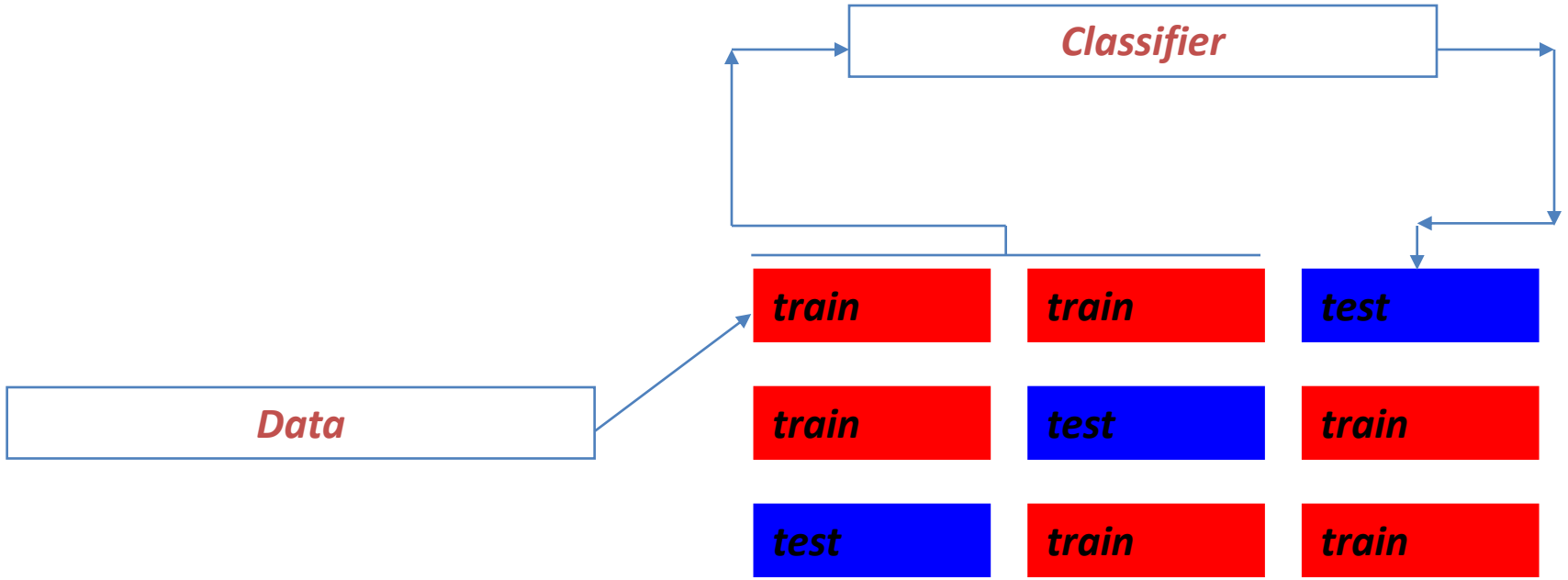
- Popular choice for k: 10 or 5
- Advantage: all available data points being used to train as well test model

K-fold cross validation



Final Accuracy = Average(Round 1, Round 2, …)

k-fold cross validation (shown for k=3)



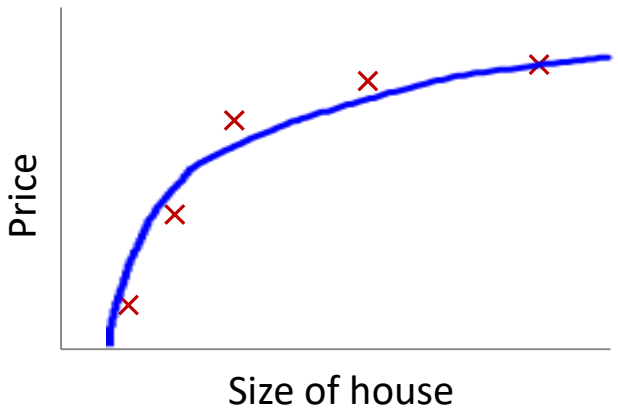
Regularization

Addressing overfitting: Two ways

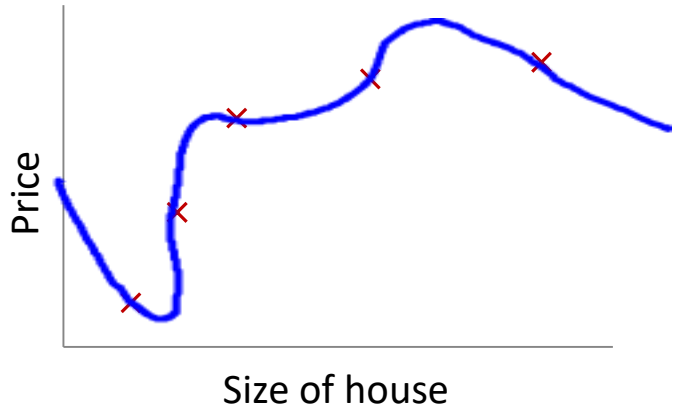
1. Reduce number of features
   - Manually select which features to keep
   - Problem: loss of some information (discarded features)

2. Regularization
   - Keep all the features, but reduce magnitude/values of parameters
   - Works well when we have a lot of features, each of which contributes a bit to predicting

Intuition of regularization



$$\theta_0 + \theta_1 x + \theta_2 x^2$$



$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Suppose we penalize and make , really small. θ_3 θ_4

$$\min_\theta \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + 1000\,\theta_3^2 + 1000\,\theta_4^2$$

Regularization for linear regression

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right]$$

$$\min_\theta J(\theta)$$

By convention, regularization is not applied on θ_0 (makes little difference to the solution)

λ: Regularization parameter

Smaller values of parameters lead to more generalizable models, less overfitting

L1 and L2 regularization

- What we are discussing is called L2 regularization or “ridge” regularization
  - adds *squared magnitude* of parameters as penalty term

- Look up L1 or “Lasso” regularization
  - adds *absolute value of magnitude* of parameters as penalty term