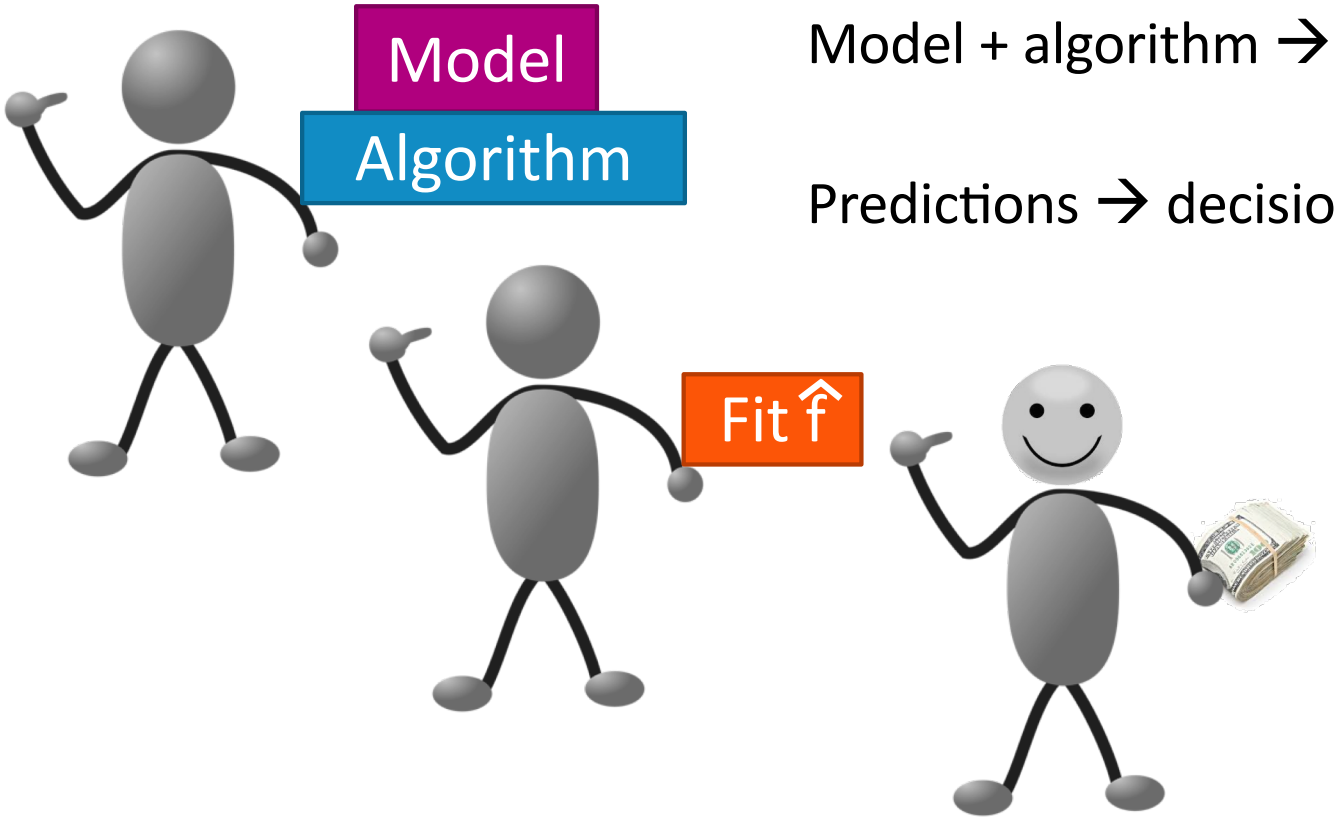
Assessing Performance

CSE 446: Machine Learning
Slides created by Emily Fox (mostly)

April 8, 2019

Make predictions, get $, right??



Model + algorithm → fitted function

Predictions → decisions → outcome

Or, how much am I losing?

How much am I losing compared to perfection?

Perfect predictions: Loss = 0

My predictions: Loss = ???

Measuring loss

Loss function:

$L(y, f_{\hat{w}}(x))$

Cost of using $\hat{w}$ at x when y is true

actual value

$\hat{f}(x)$ = predicted value $\hat{y}$

Examples:

Squared error: $L(y, f_{\hat{w}}(x)) = (y - f_{\hat{w}}(x))^2$

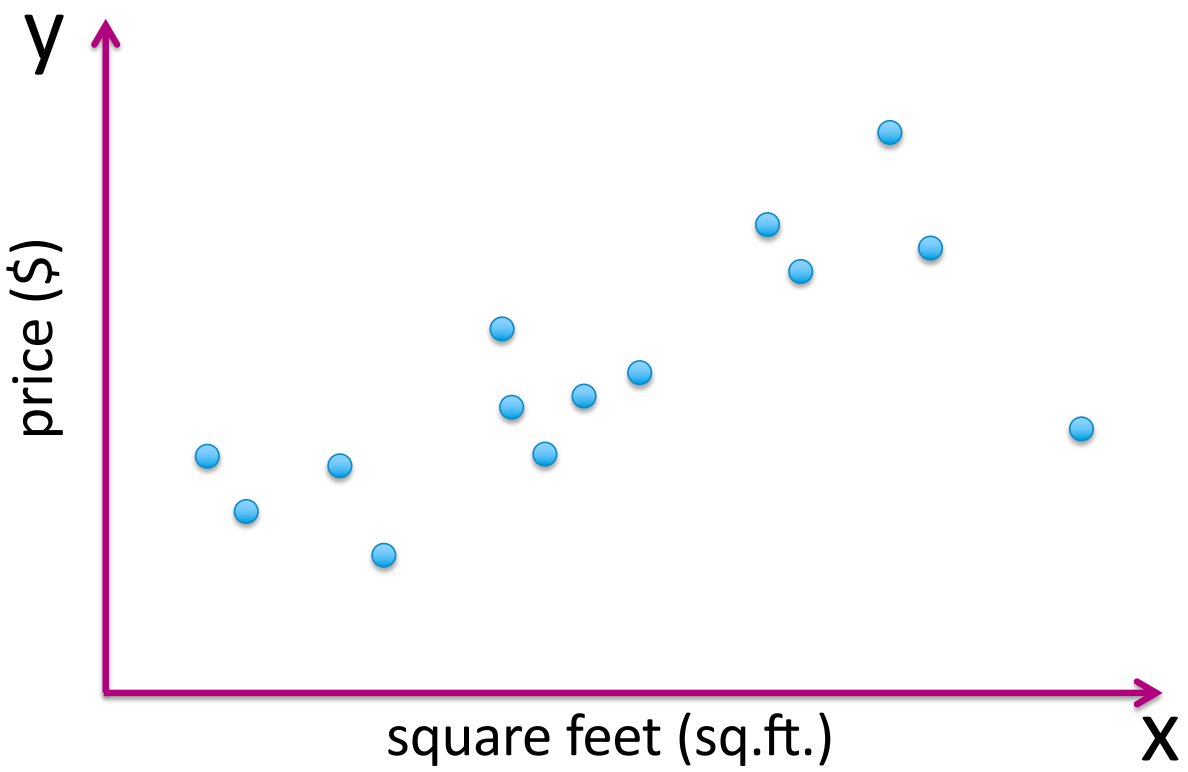
Absolute error: $L(y, f_{\hat{w}}(x)) = |y - f_{\hat{w}}(x)|$

"Remember that all models are wrong; the practical question is how wrong do they have to be to not be useful." George Box, 1987.

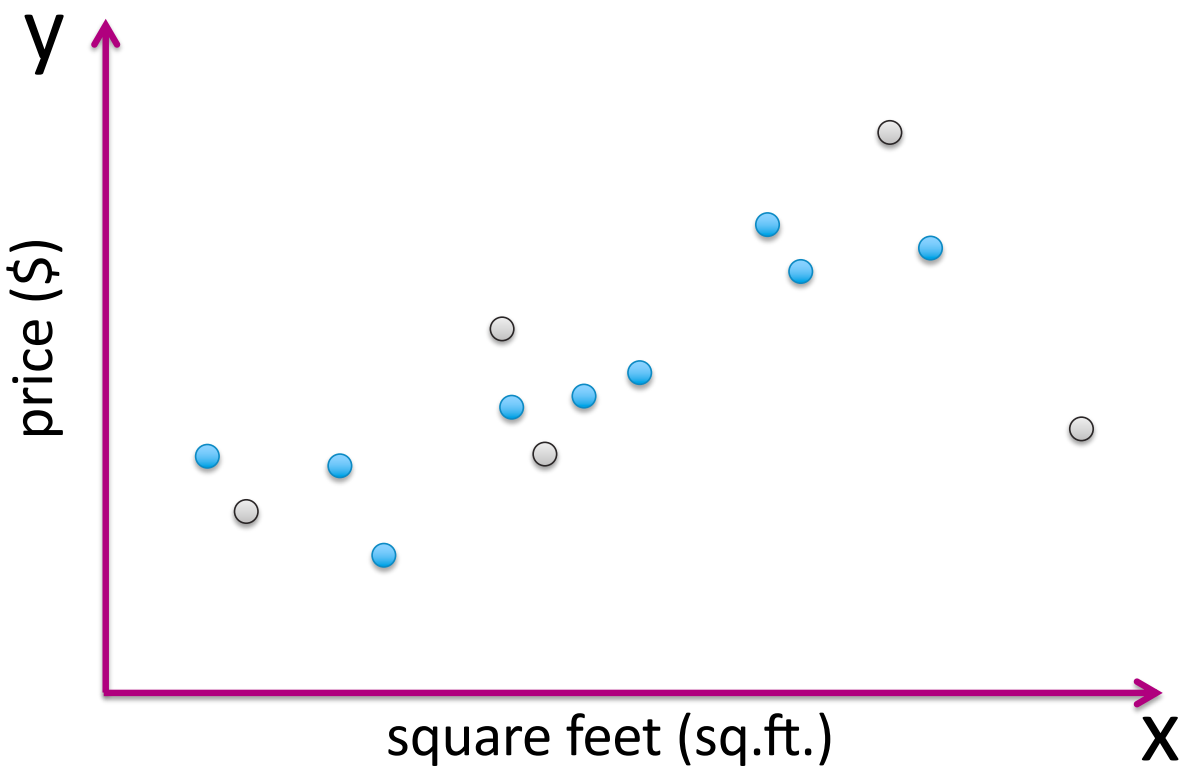
Assessing the loss

Assessing the loss
Part 1: Training error

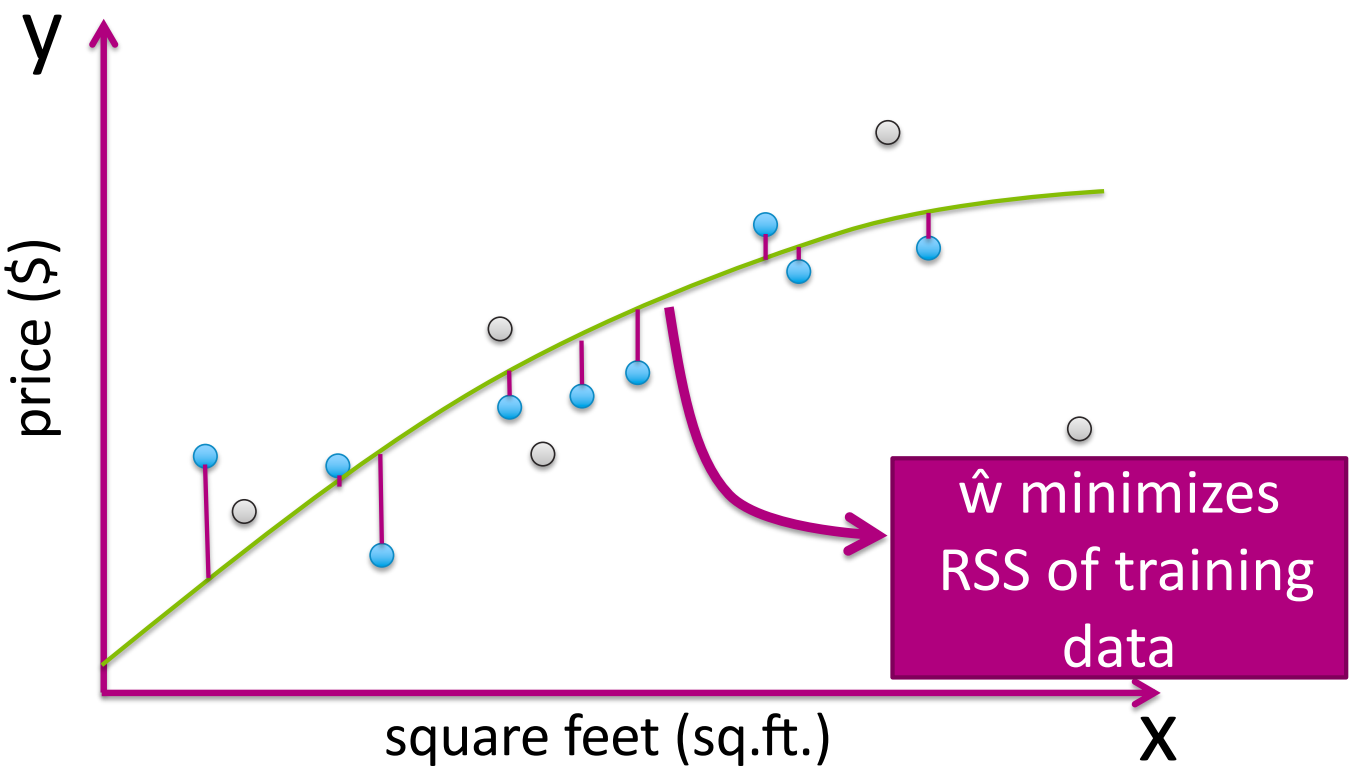
Define training data



Define training data



Example:
Fit quadratic to minimize RSS



Compute training error

1. Define a loss function $L(y, f_{\hat{w}}(x))$
   - E.g., squared error, absolute error,…

2. Training error using $f_{\hat{w}}$

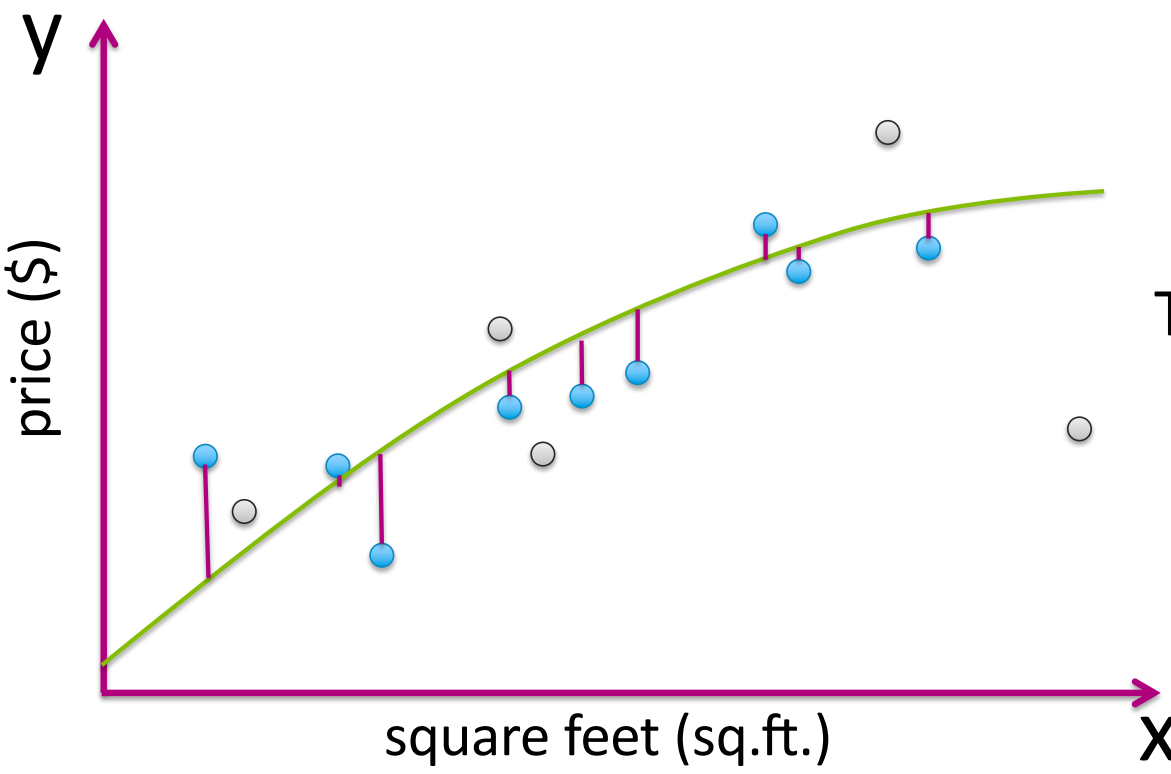
   = avg. loss on houses in training set

   = $\frac{1}{n}\sum_{i=1}^{n} L(y_i, f_{\hat{w}}(x_i))$

fit using training data

Example:

Use squared error loss $(y-f_{\hat{w}}(x))^2$



Training error ($\hat{w}$) = 1/n *

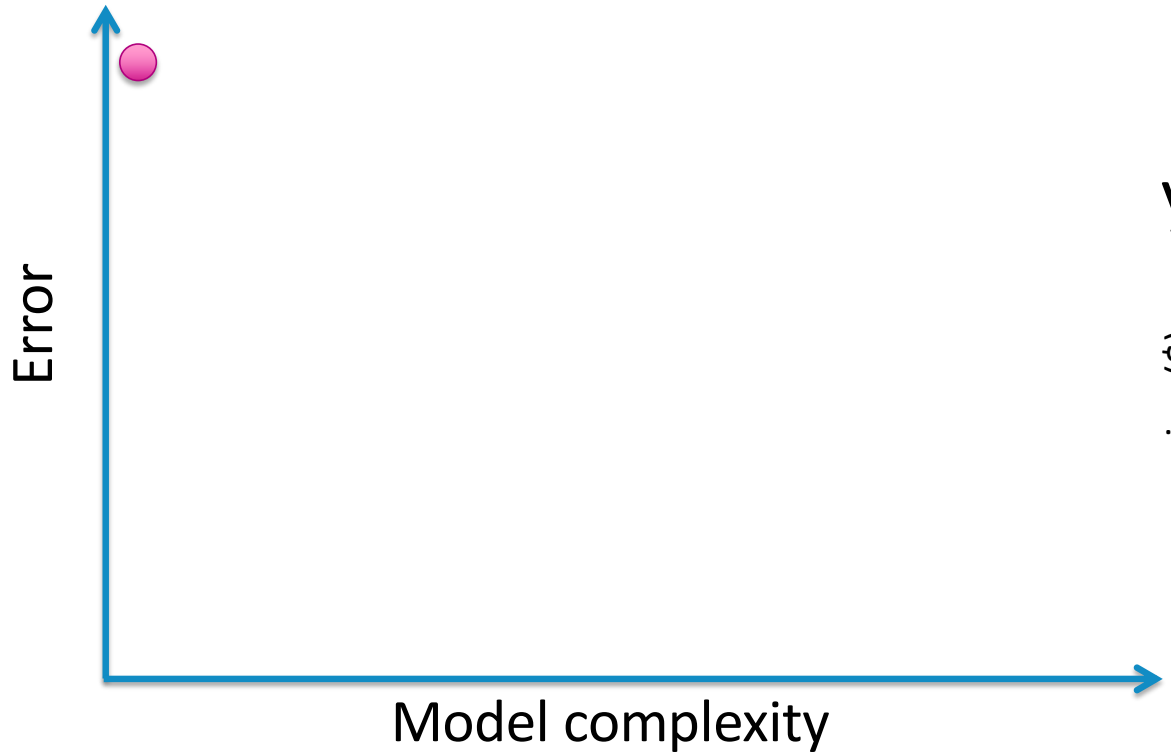
$[(\$_{train\ 1}-f_{\hat{w}}(sq.ft._{train\ 1}))^2$

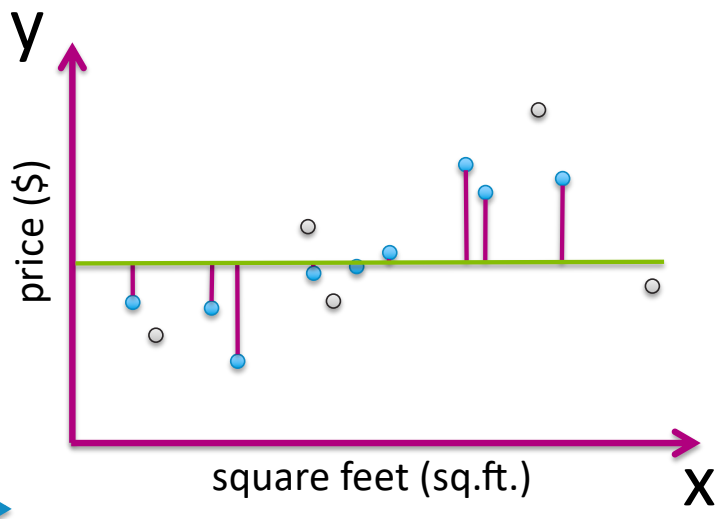
$+ (\$_{train\ 2}-f_{\hat{w}}(sq.ft._{train\ 2}))^2$

$+ (\$_{train\ 3}-f_{\hat{w}}(sq.ft._{train\ 3}))^2$

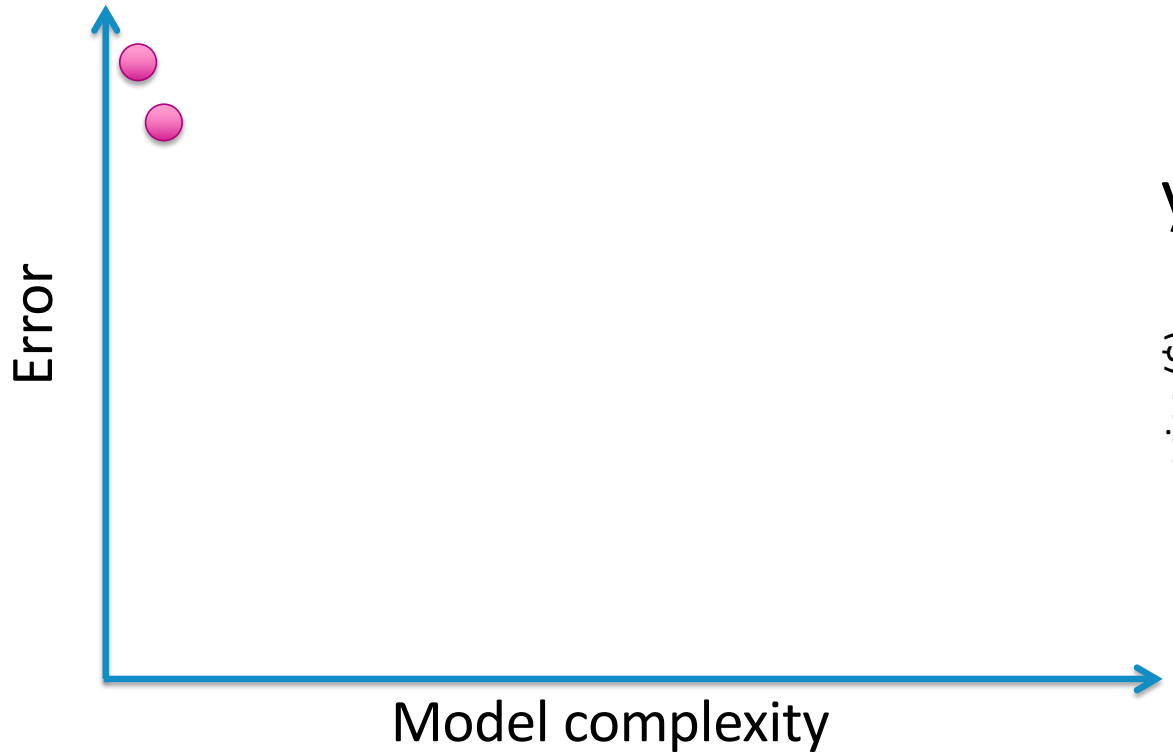
+ … include all training houses]

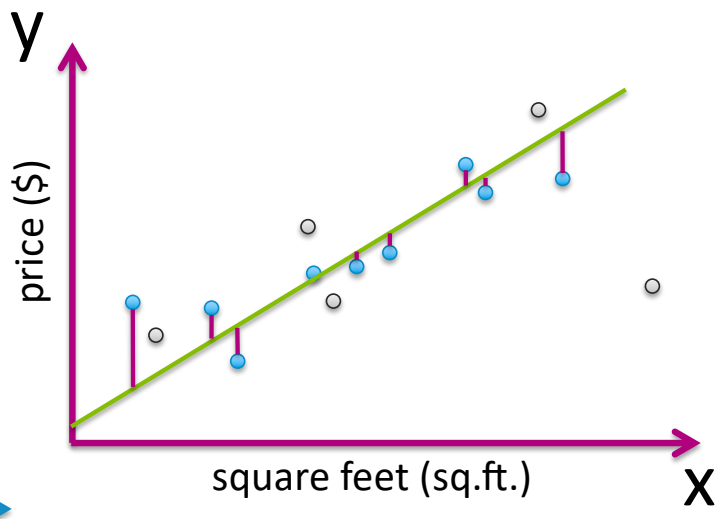
Training error vs. model complexity



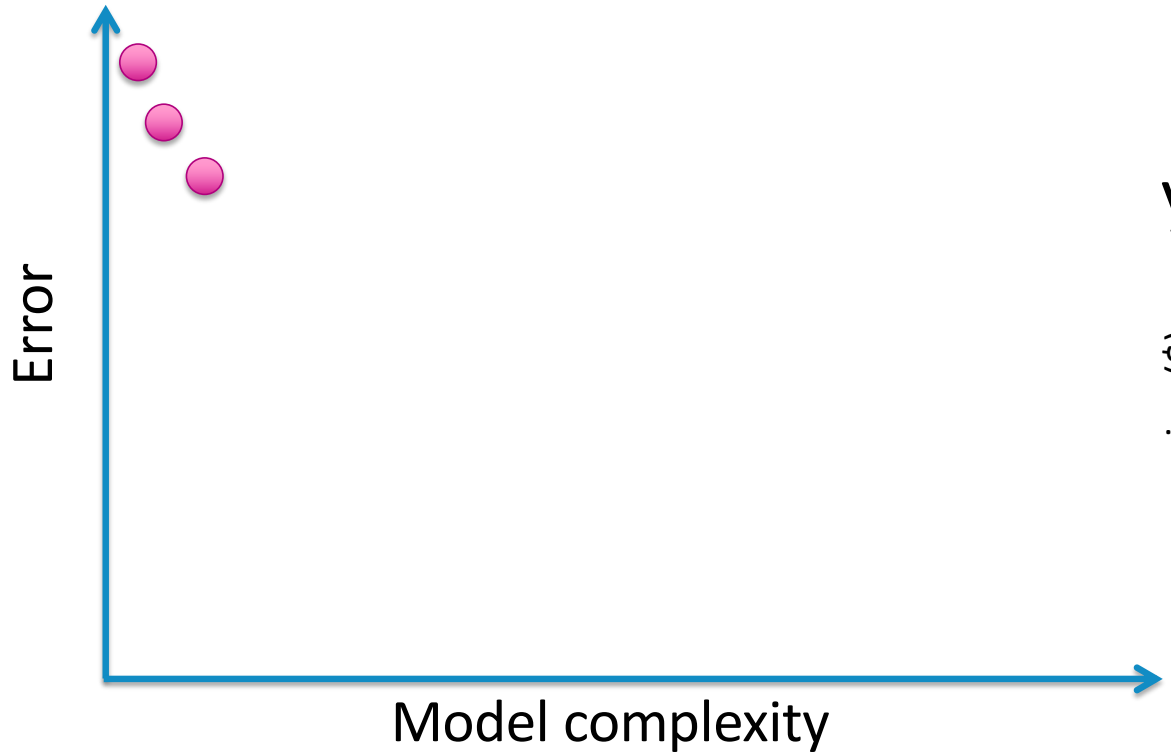


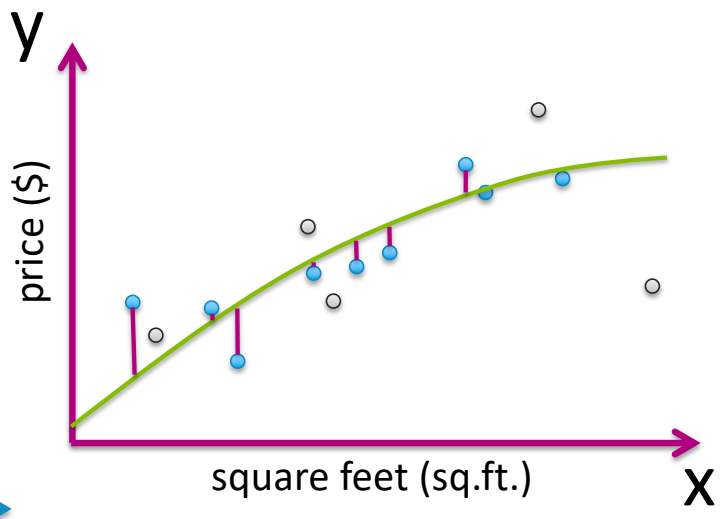
Training error vs. model complexity



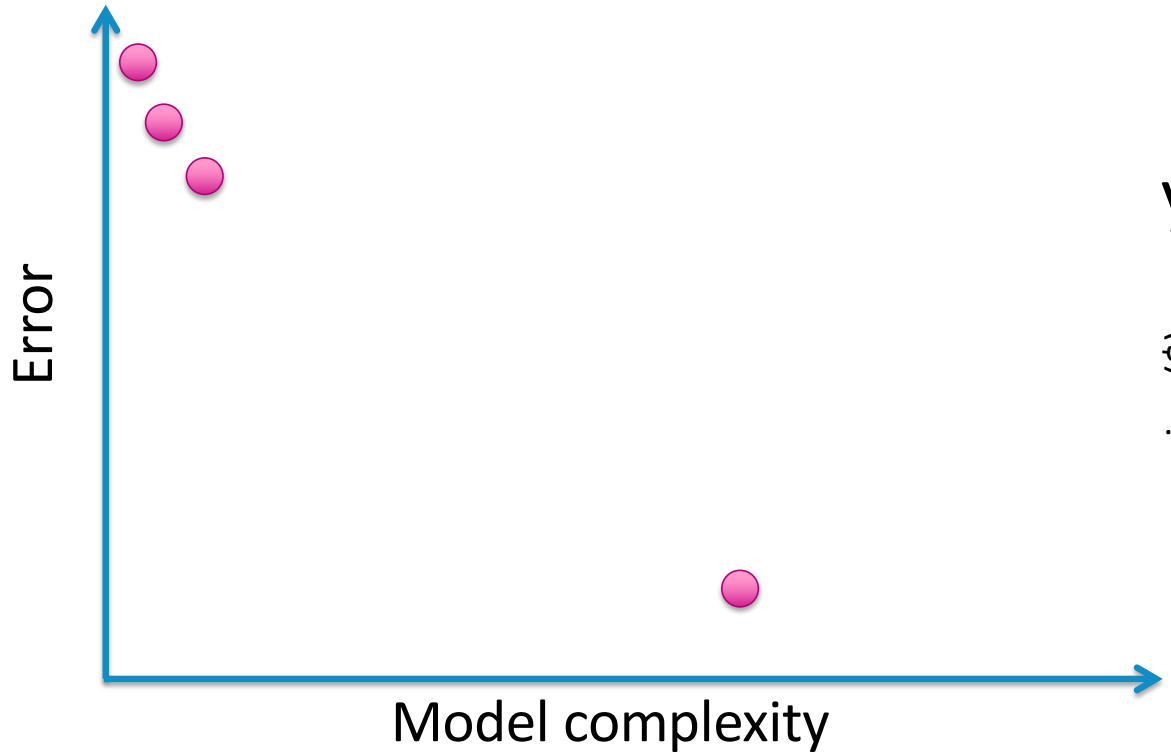


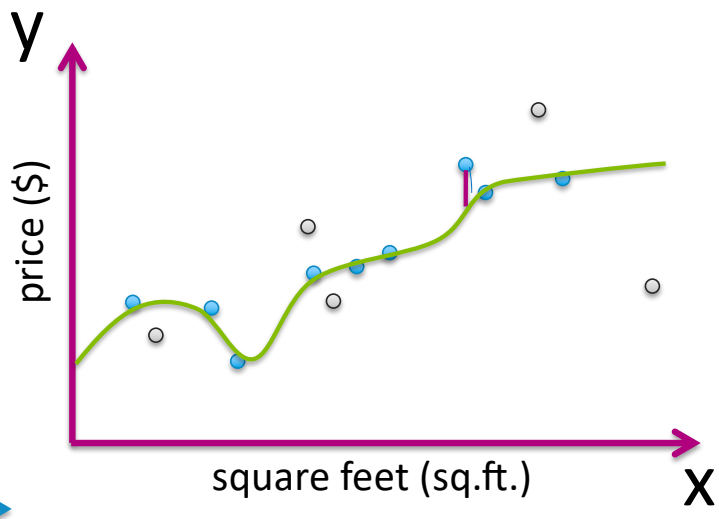
Training error vs. model complexity



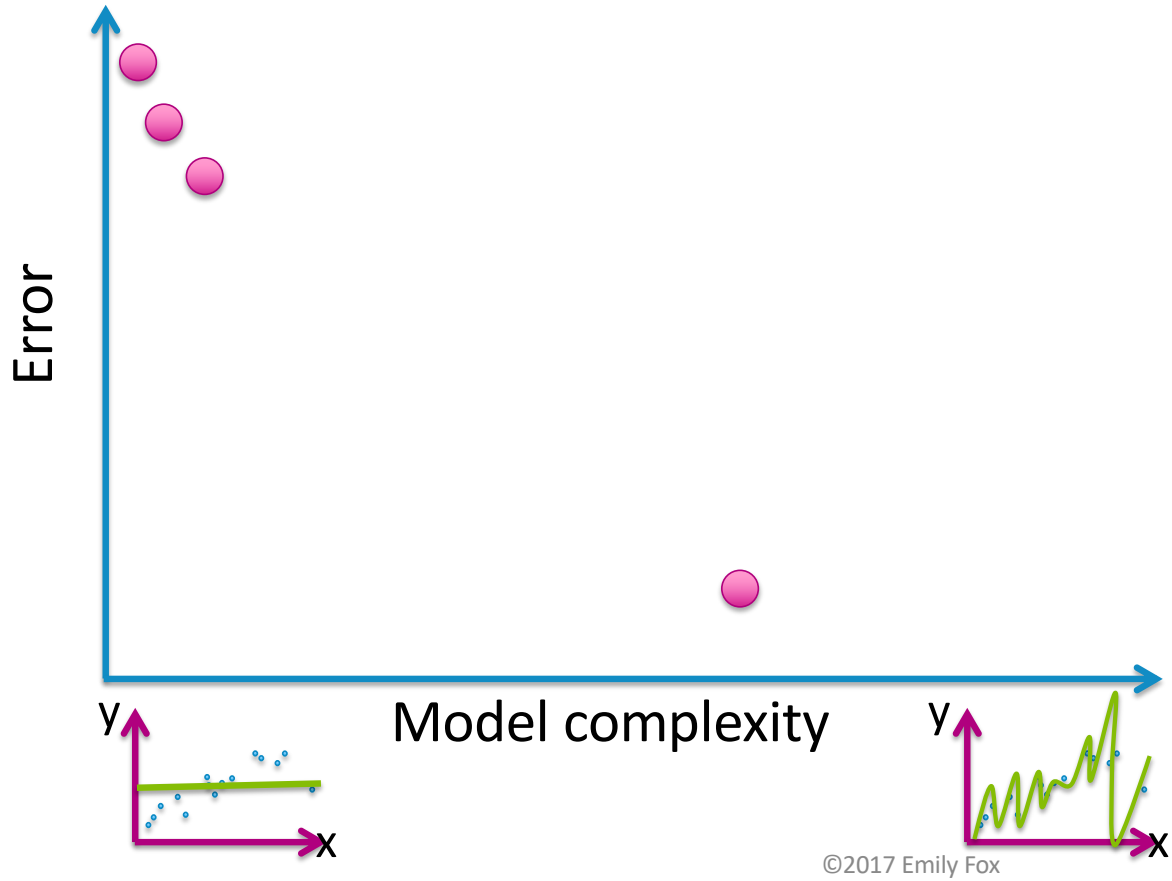


Training error vs. model complexity



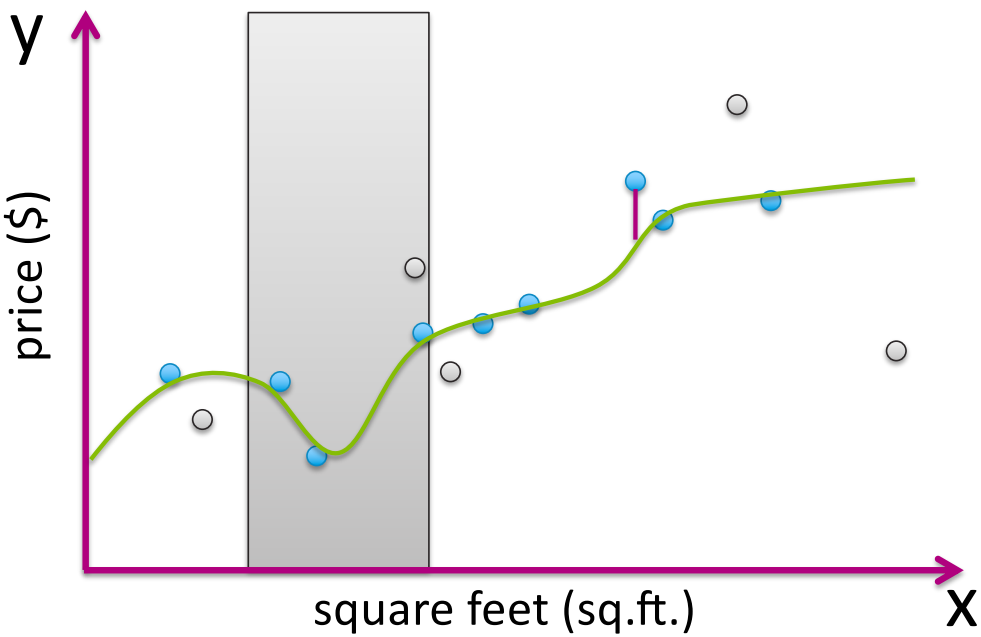


Training error vs. model complexity



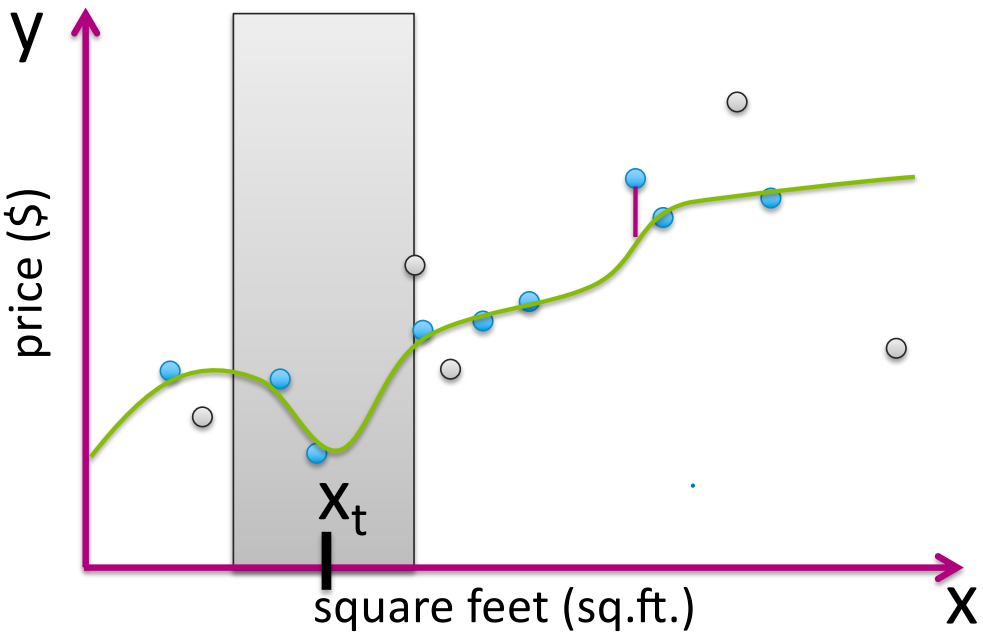
Is training error a good measure of predictive performance?

How do we expect to perform on a new house?



Is training error a good measure of predictive performance?

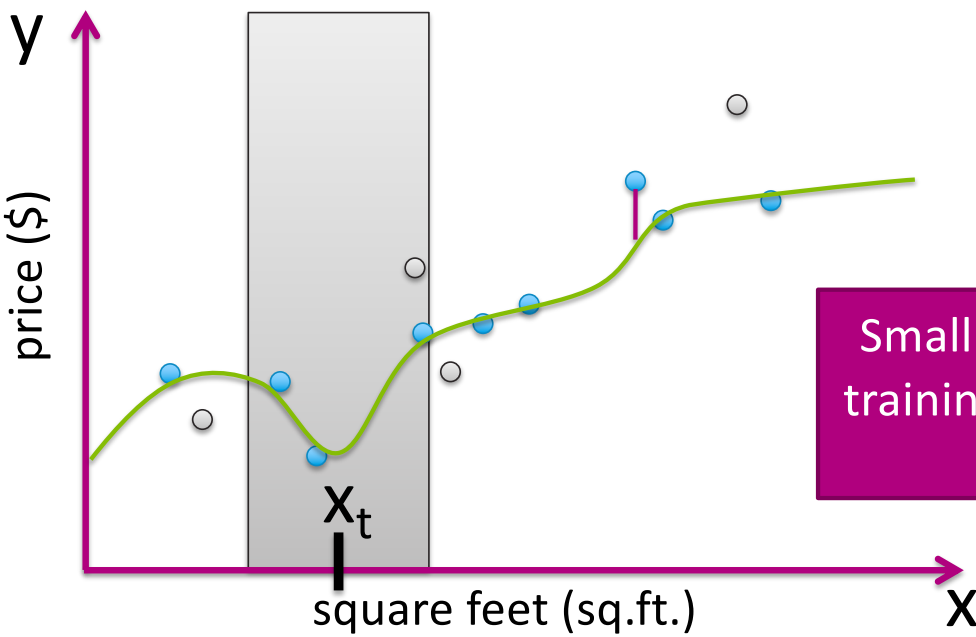
Is there something particularly bad about having x_t sq.ft.??



Is training error a good measure of predictive performance?

Issue:

Training error is overly optimistic…$\hat{w}$ was fit to training data

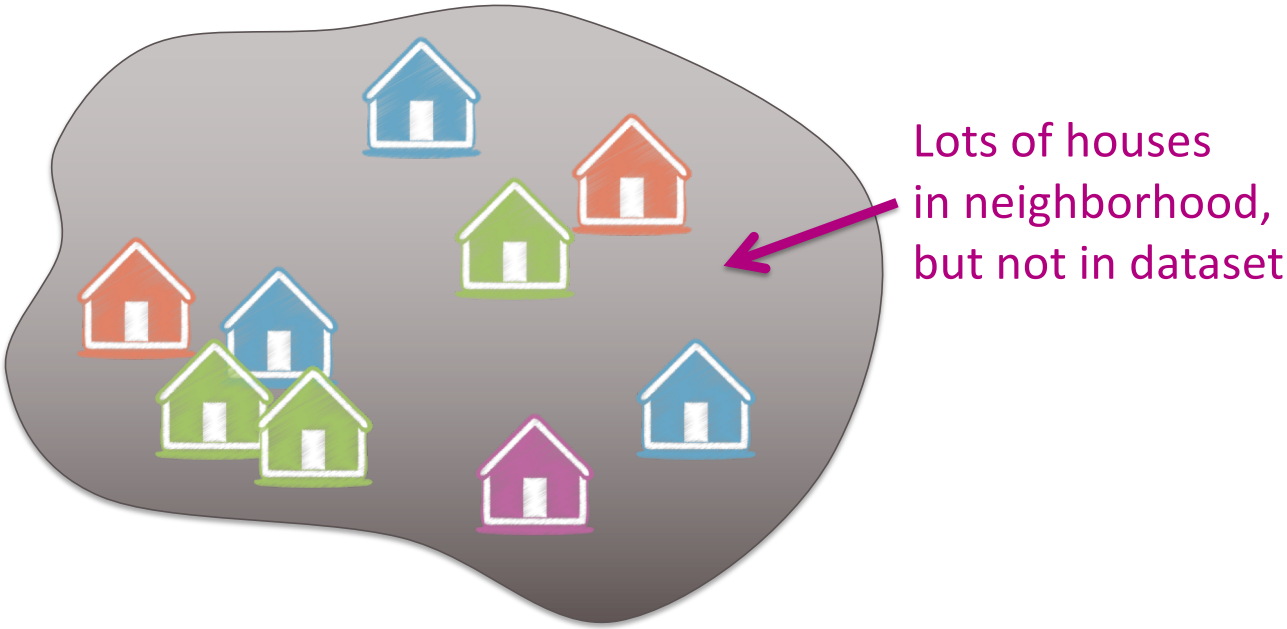


Small training error ≠> good predictions unless training data includes everything you might ever see

Assessing the loss
Part 2: Generalization (true) error

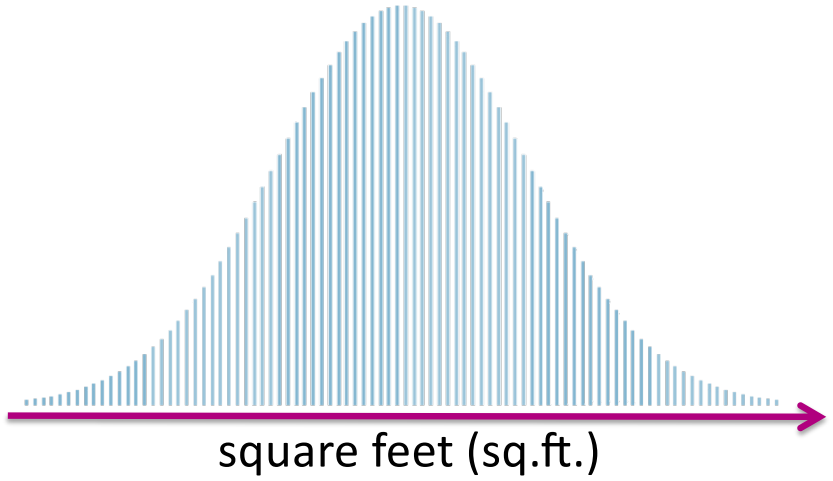
Generalization error

Really want estimate of loss over all possible (🏠,$) pairs



Distribution over houses

In our neighborhood, houses of what # sq.ft. ( ) are we likely to see?



Distribution over sales prices

For houses with a given # sq.ft. (), what house prices $ are we likely to see?

Generalization error definition

Really want estimate of loss over all possible ( ,$) pairs

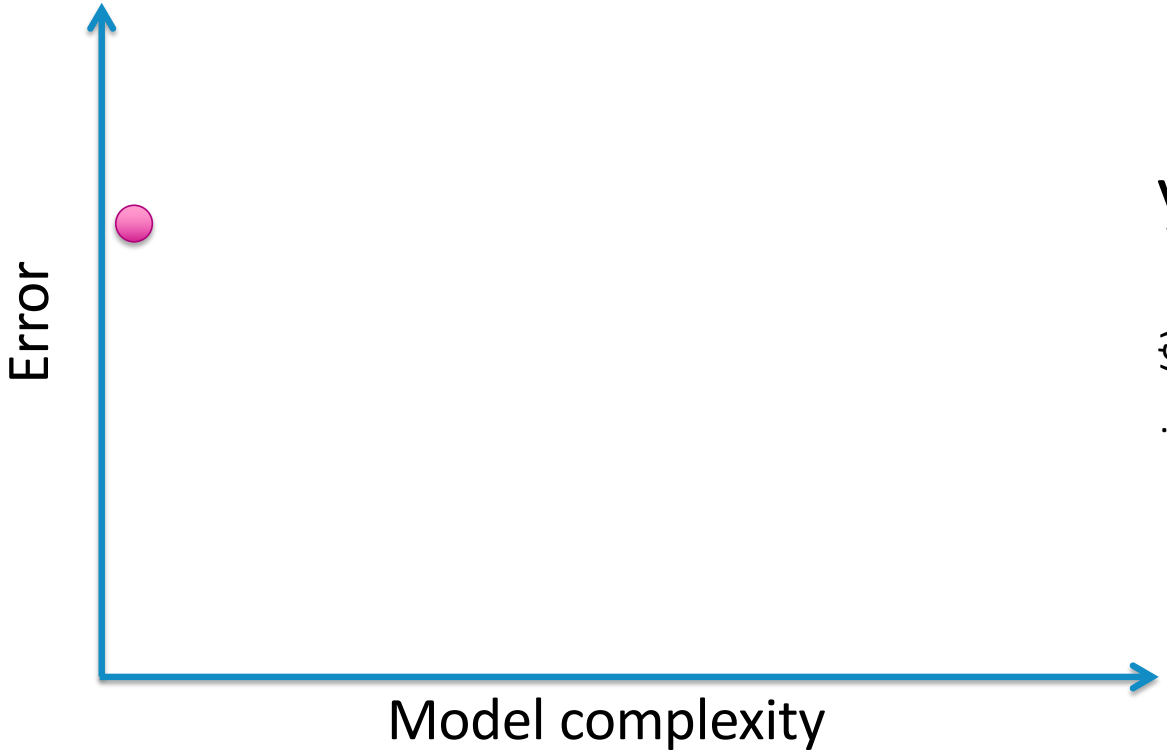
average over all possible (x,y) pairs weighted by how likely each is

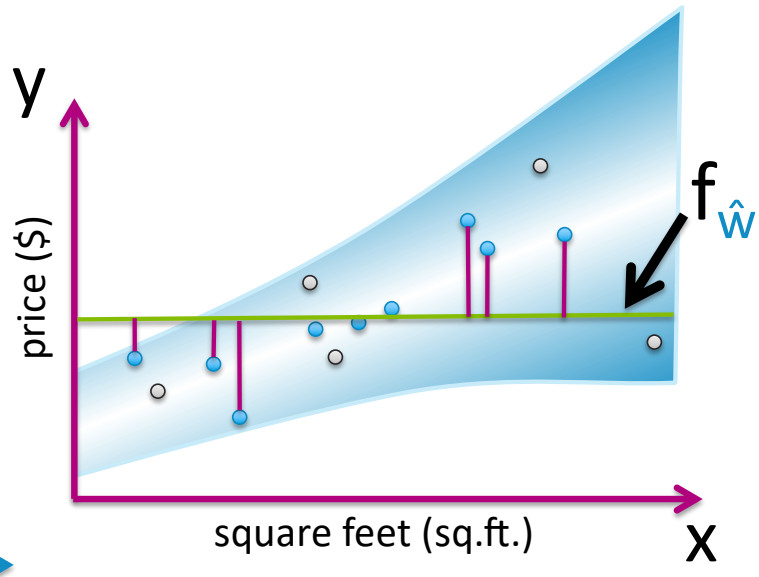
Formally:

$$\text{generalization error} = E_{x,y}\left[L(y, f_{\hat{w}}(x))\right]$$

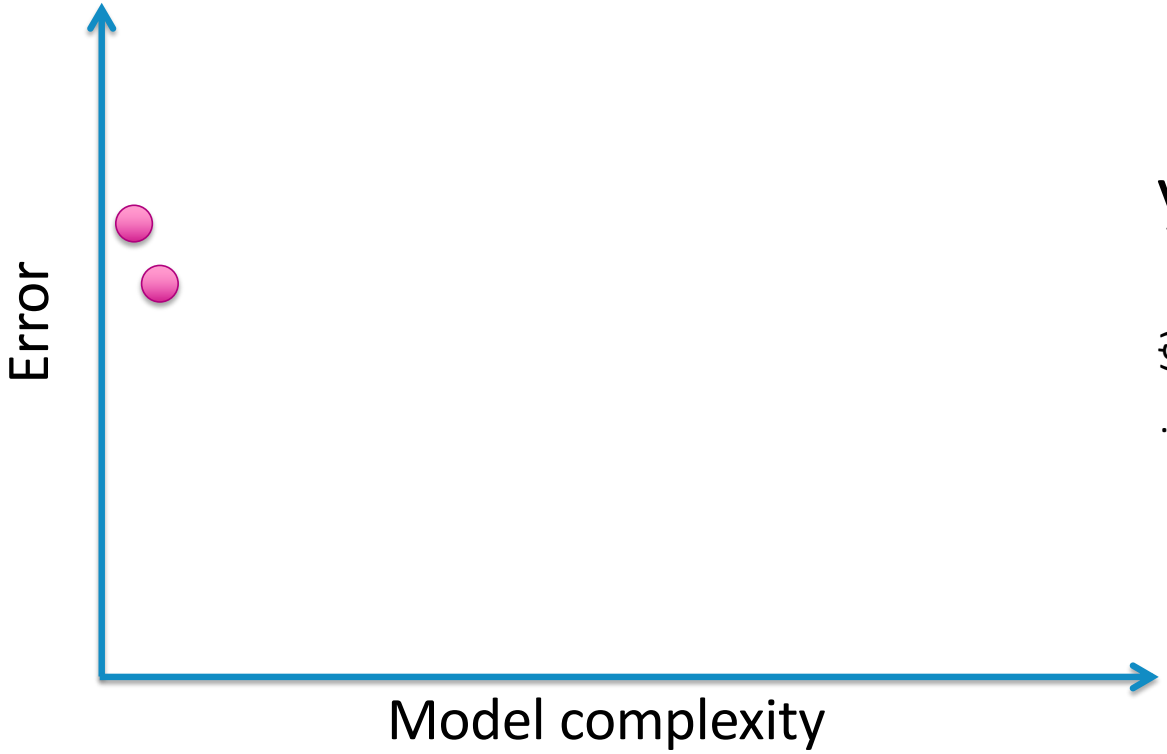
fit using training data

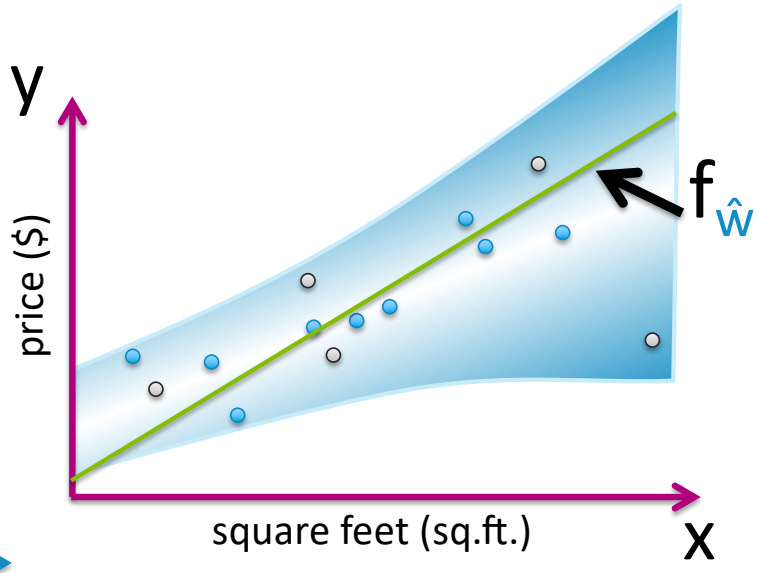
Generalization error vs. model complexity



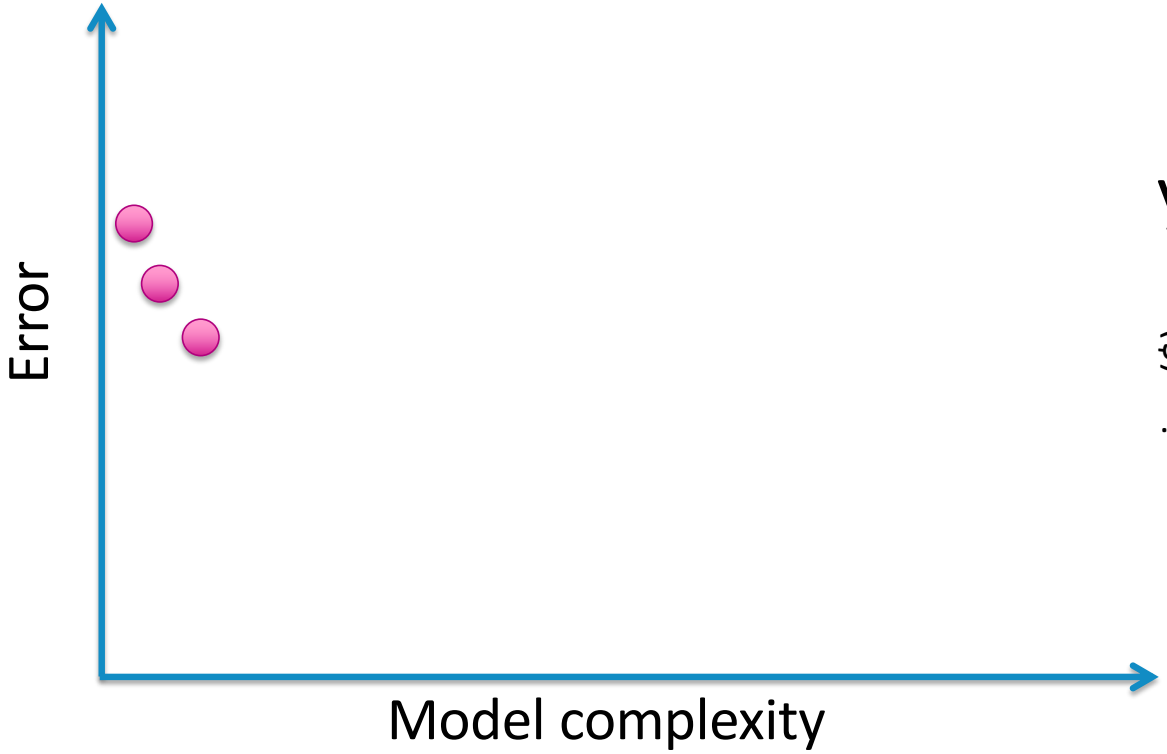


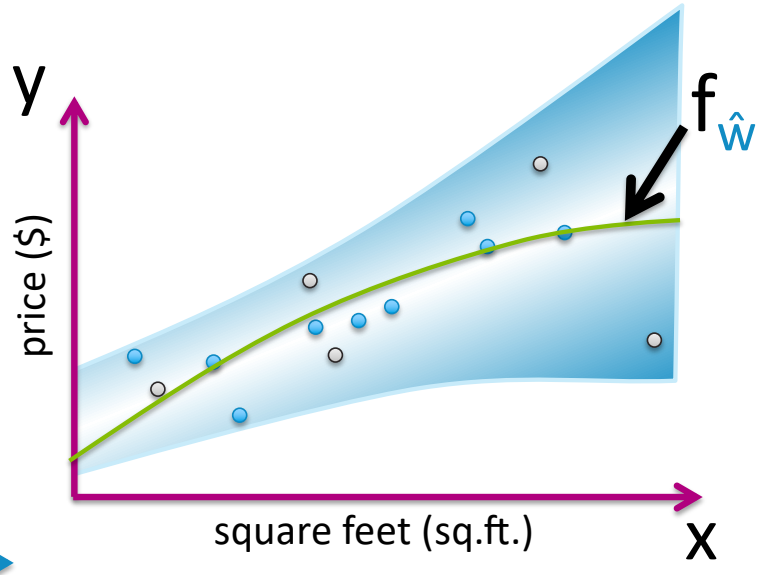
Generalization error vs. model complexity



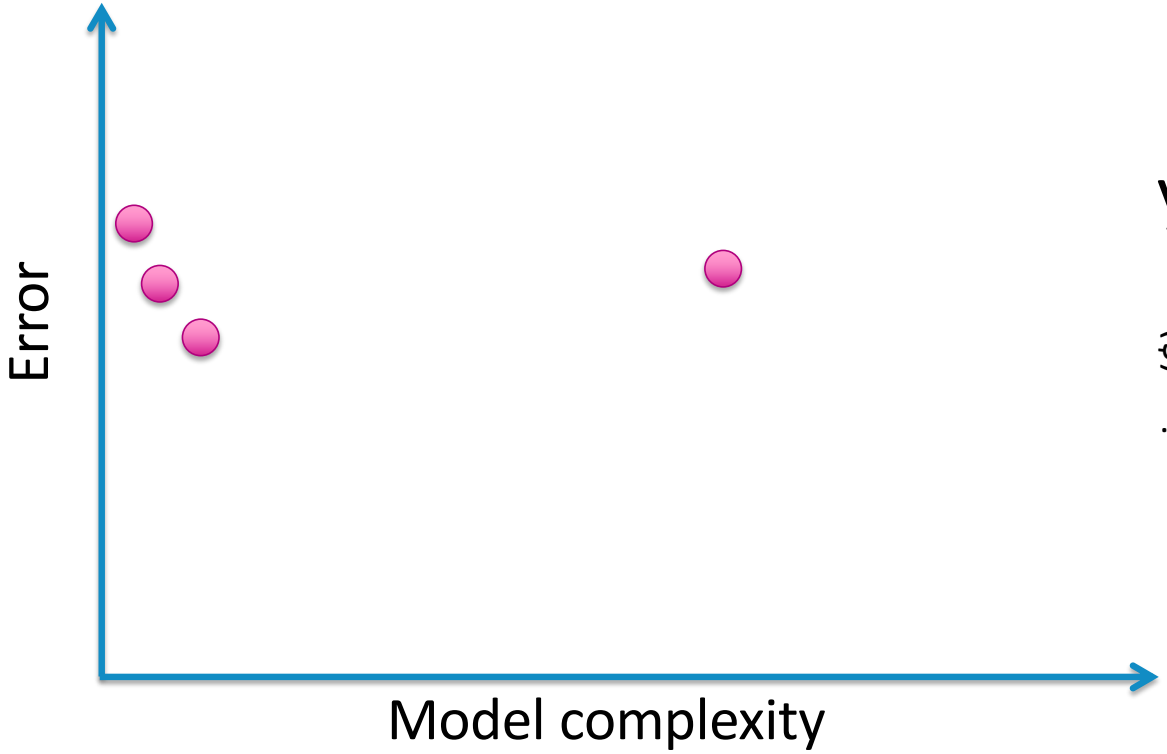


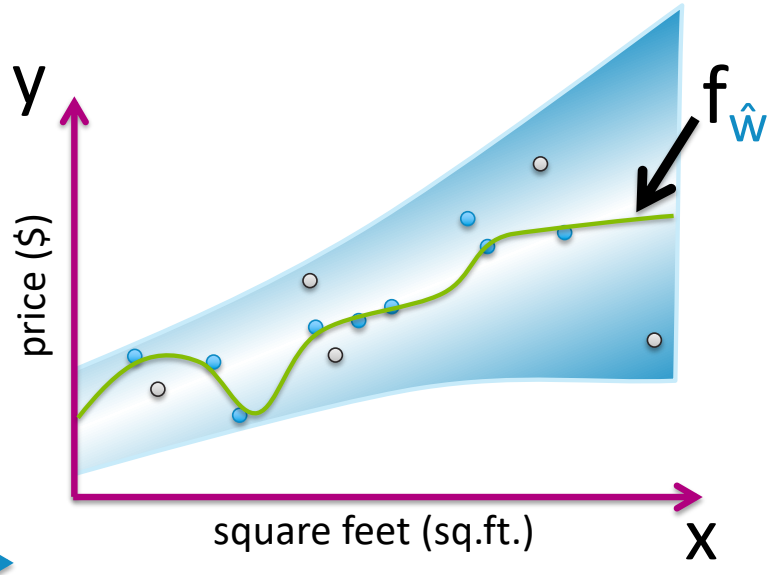
Generalization error vs. model complexity



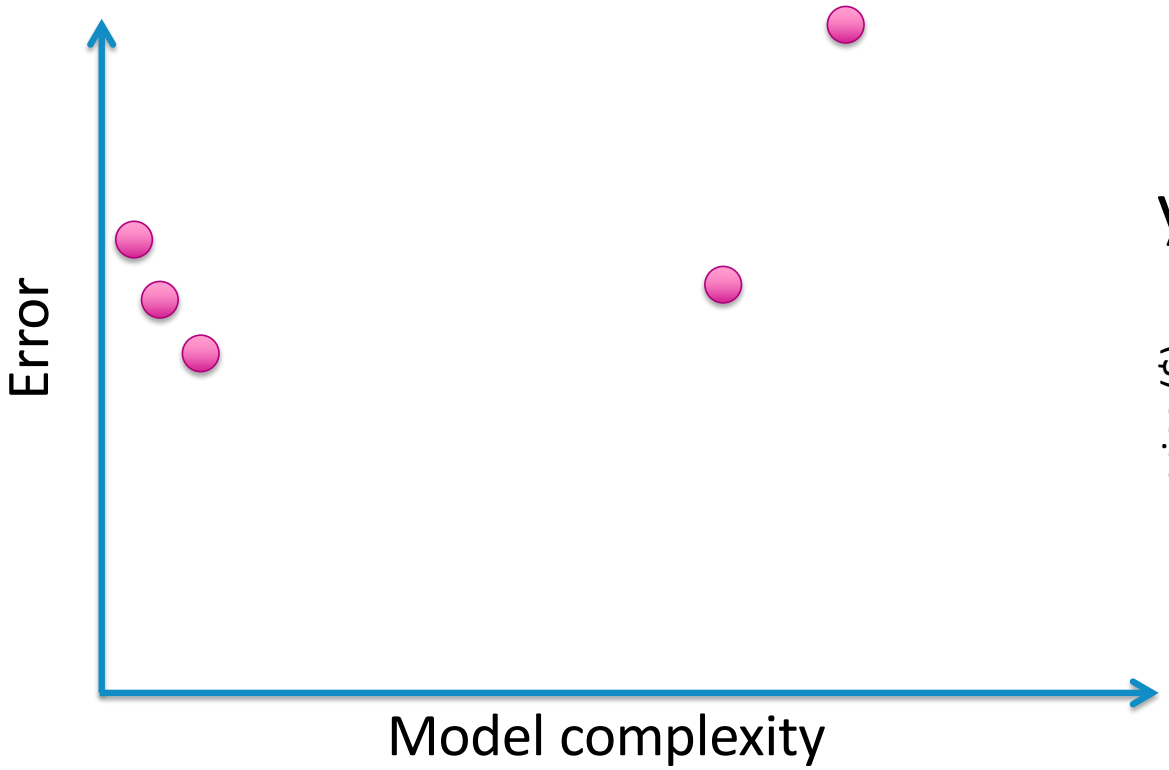


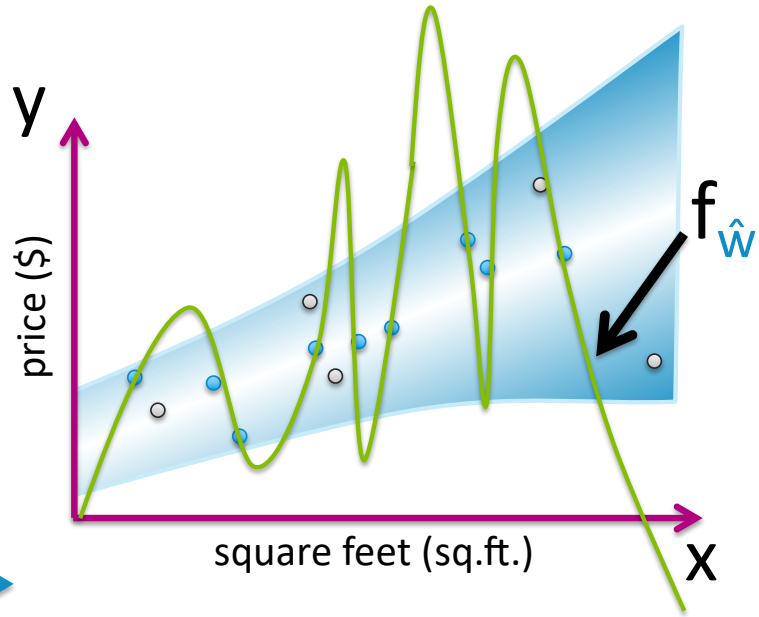
Generalization error vs. model complexity



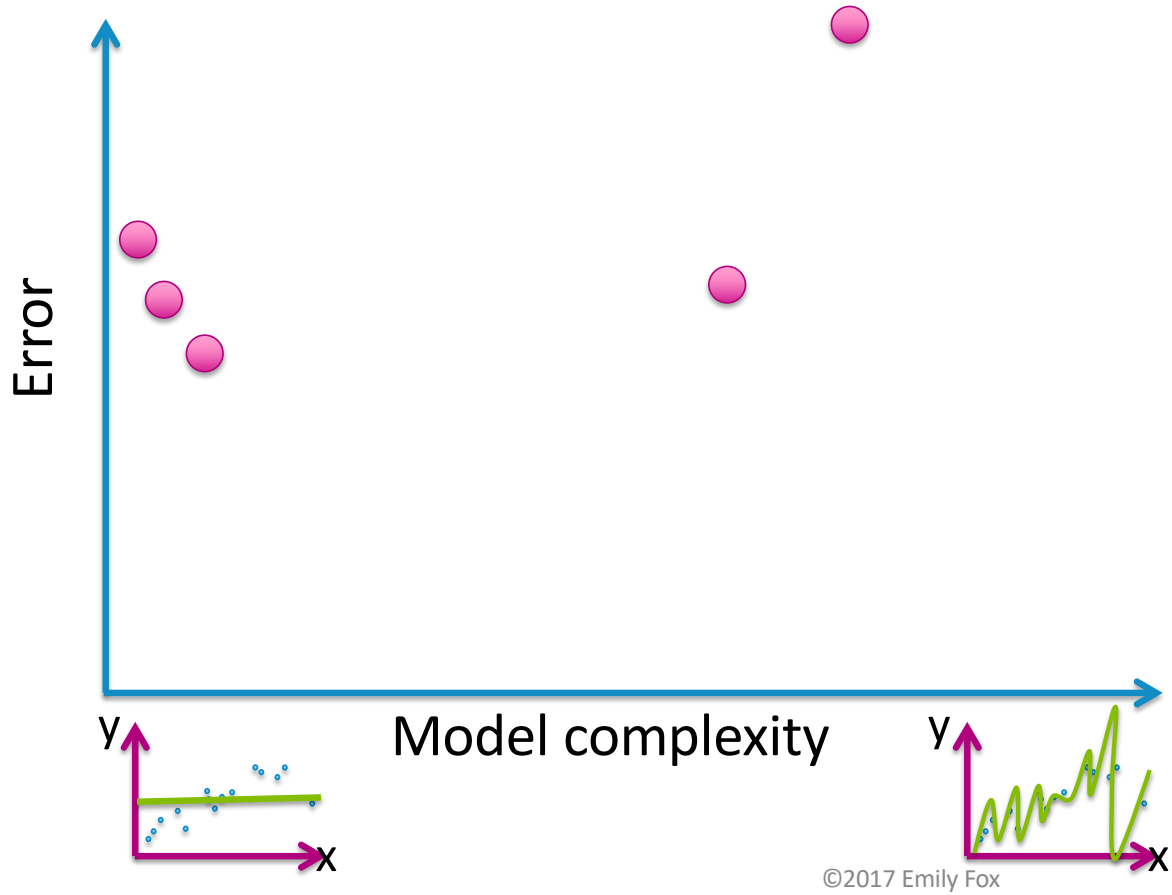


Generalization error vs. model complexity





Generalization error vs. model complexity



Can't compute!

Assessing the loss
Part 3: Test error

Approximating generalization error

Wanted estimate of loss over all possible (🏠,$) pairs

Approximate by looking at houses not in training set

Forming a test set

Hold out some (,$) that are *not* used for fitting the model

Training set

Test set

Forming a test set

Hold out some (🏠,$) that are *not* used for fitting the model

Proxy for "everything you might see"

Test set

Compute test error

Test error

= avg. loss on houses in test set

$$= \frac{1}{N_{test}} \sum_{i \text{ in test set}} L(y_i, f_{\hat{w}}(x_i))$$

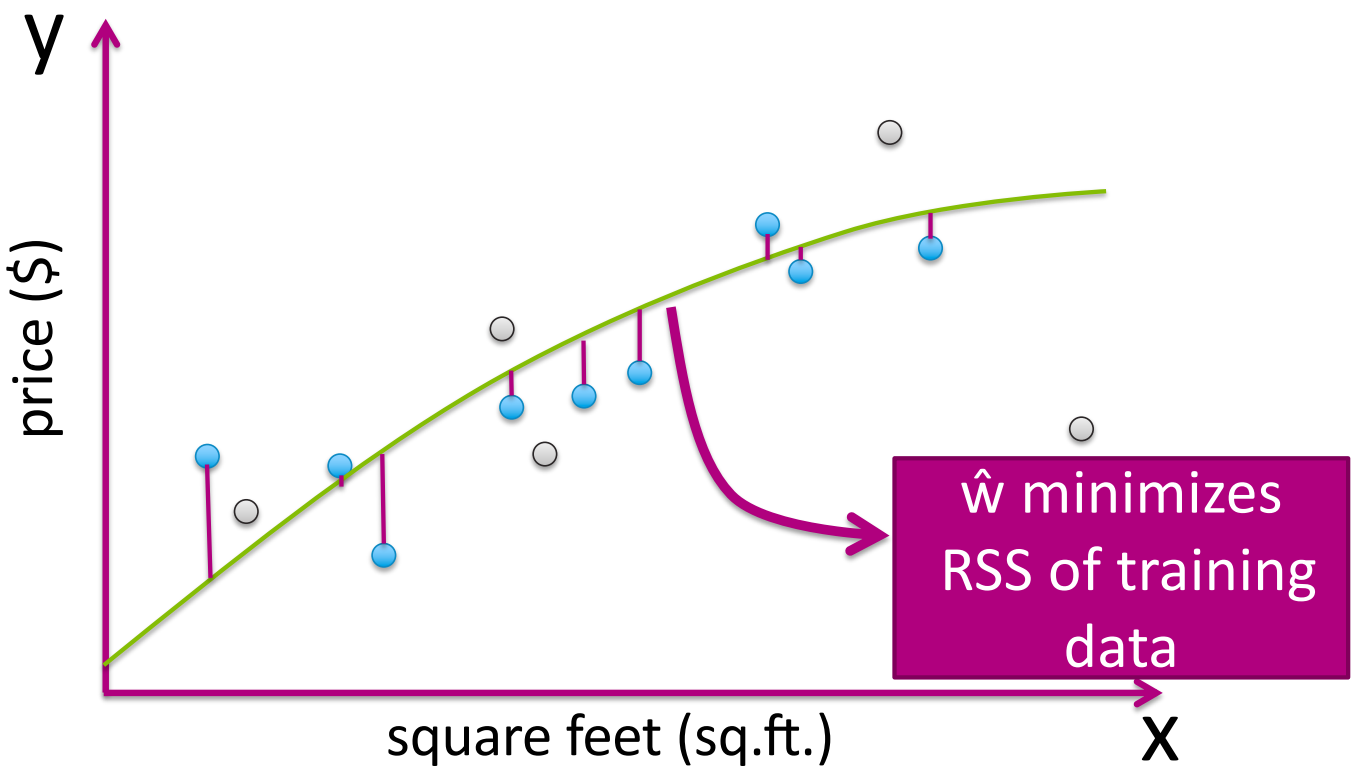
test points

fit using training data

has never seen test data!

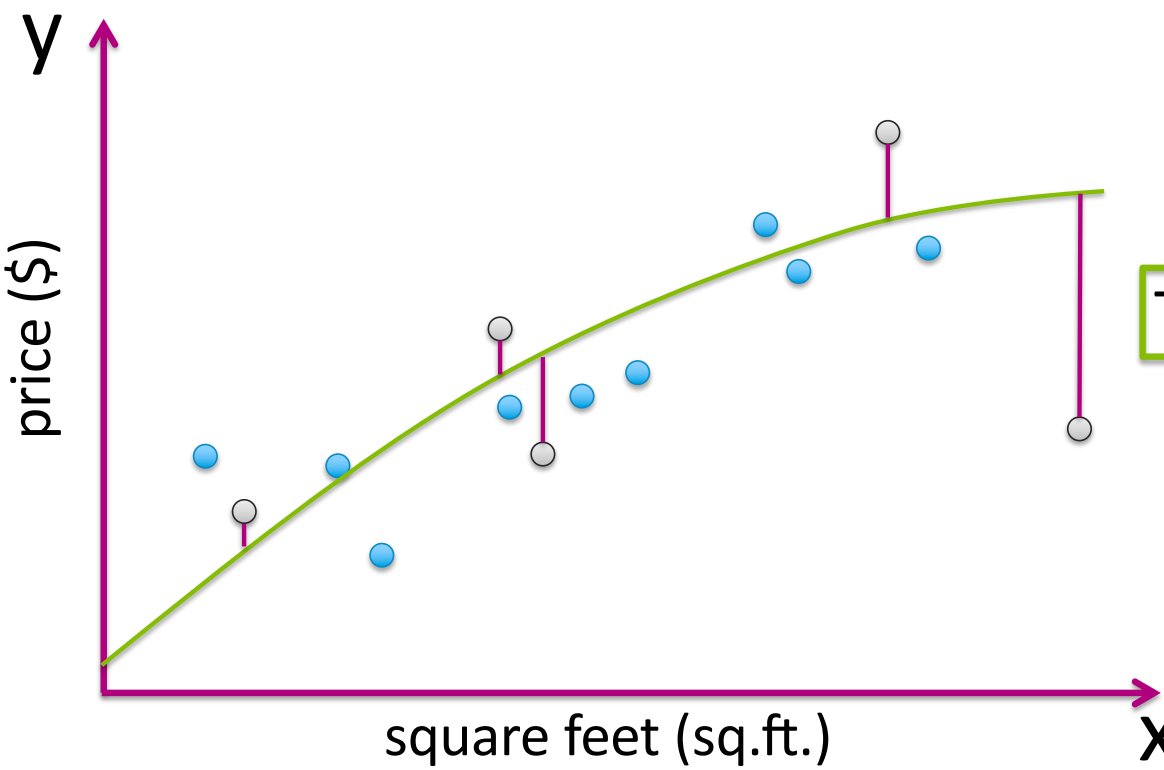
Example:

As before, fit quadratic to training data



Example:

As before, use squared error loss $(y-f_{\hat{w}}(x))^2$



Test error ($\hat{w}$) = $1/N_{test}$ *

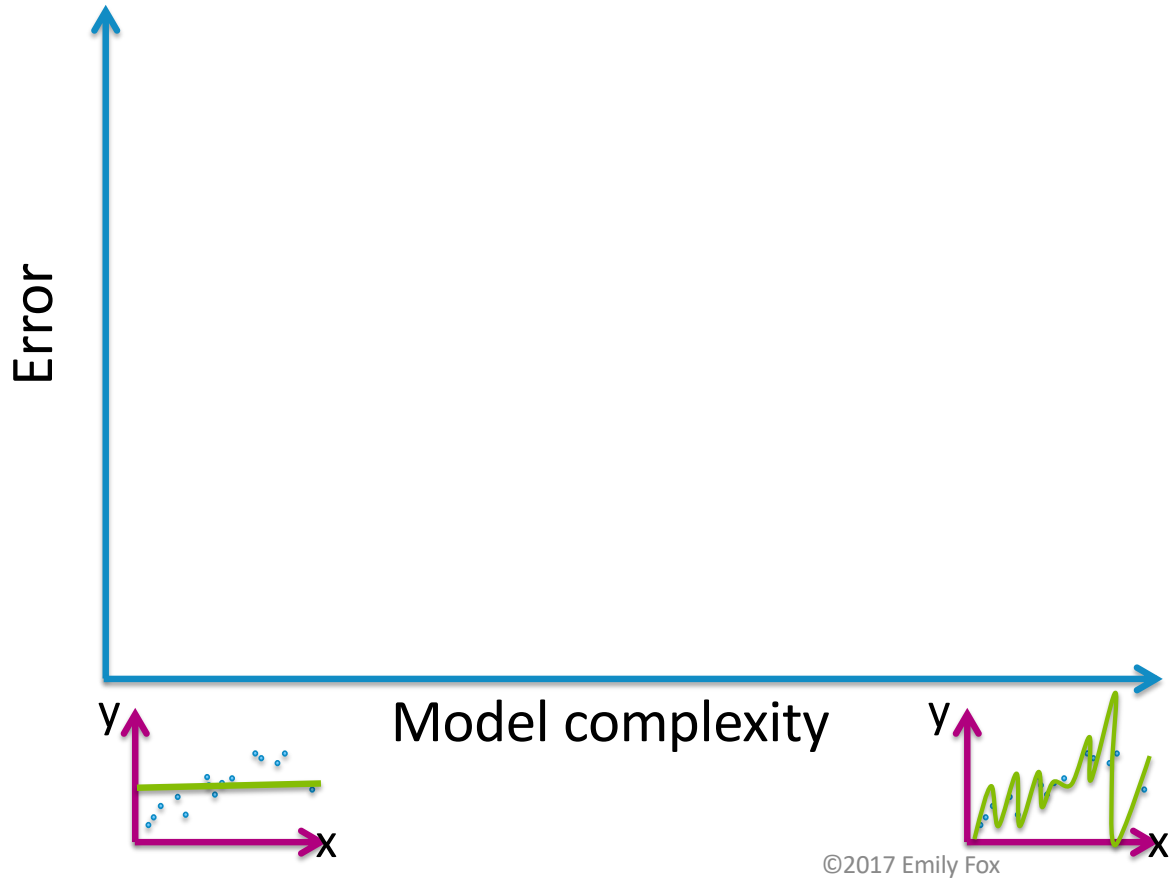
$[(\$_{test\ 1}-f_{\hat{w}}(sq.ft._{test\ 1}))^2$

$+ (\$_{test\ 2}-f_{\hat{w}}(sq.ft._{test\ 2}))^2$

$+ (\$_{test\ 3}-f_{\hat{w}}(sq.ft._{test\ 3}))^2$

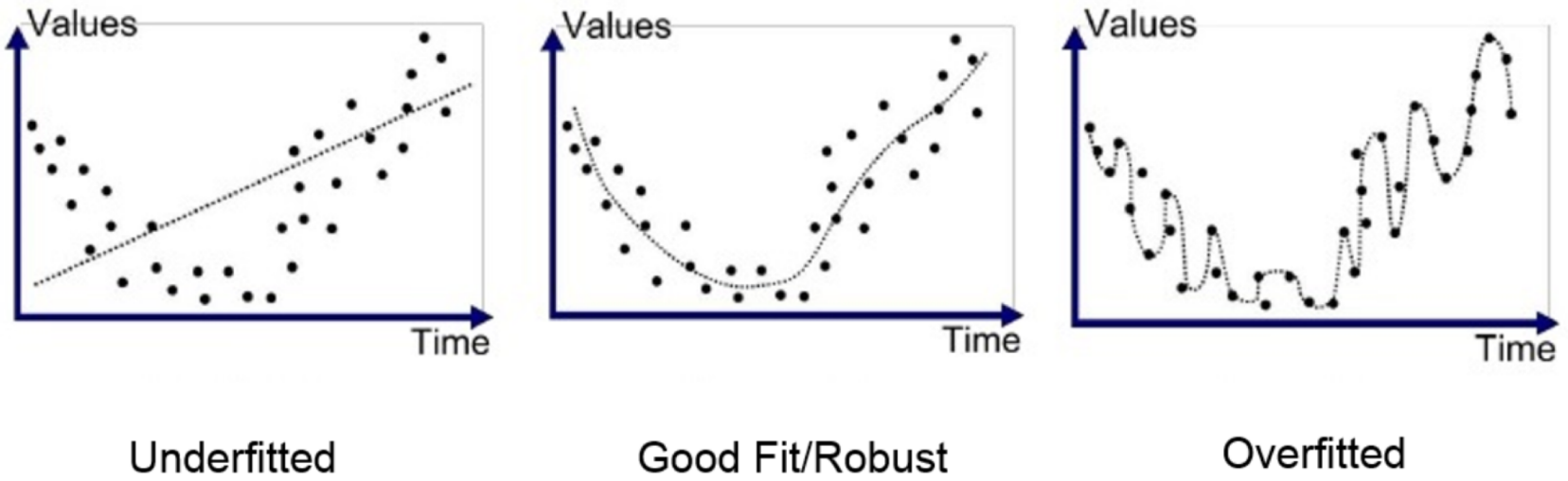
+ … include all test houses]

Training, true, & test error vs. model complexity



Overfitting if:

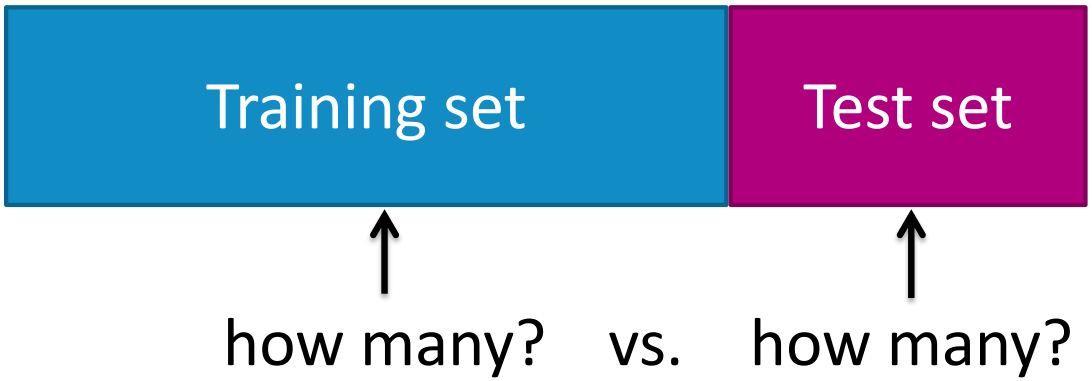
Underfitting vs Overfitting



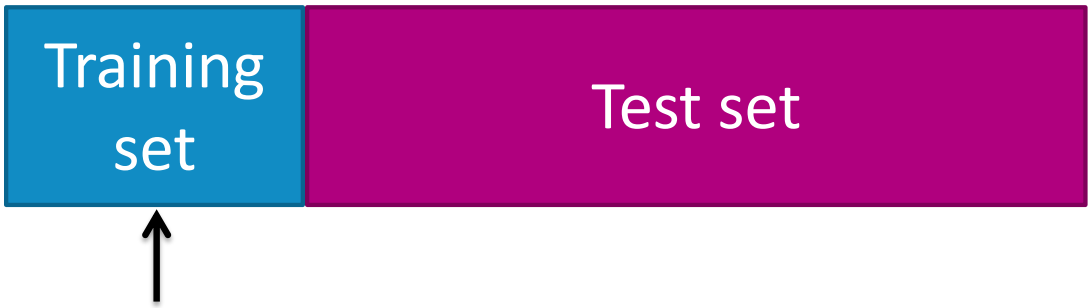
Courtesy Blog@AlgoTrading101

Training/test split

Training/test splits

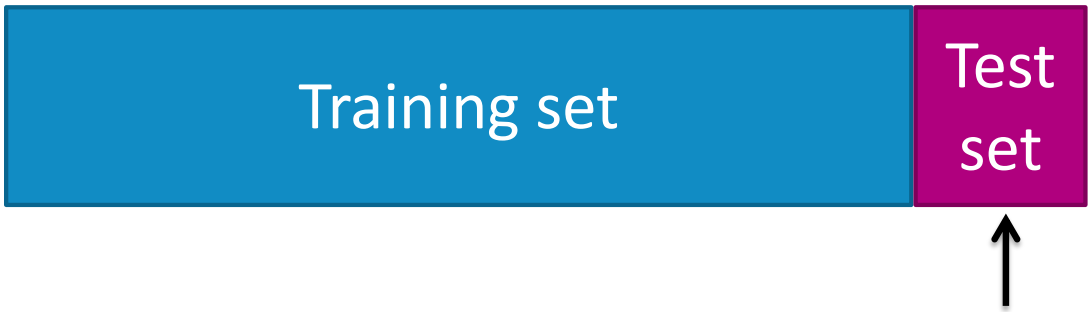


Training/test splits



Too few → $\hat{w}$ poorly estimated

Training/test splits



Too few → test error bad approximation of generalization error

Training/test splits

Typically, just enough test points to form a reasonable estimate of generalization error

If this leaves too few for training, other methods like cross validation (will see later…)

3 sources of error +
the bias-variance tradeoff

3 sources of error

In forming predictions, there are 3 sources of error:

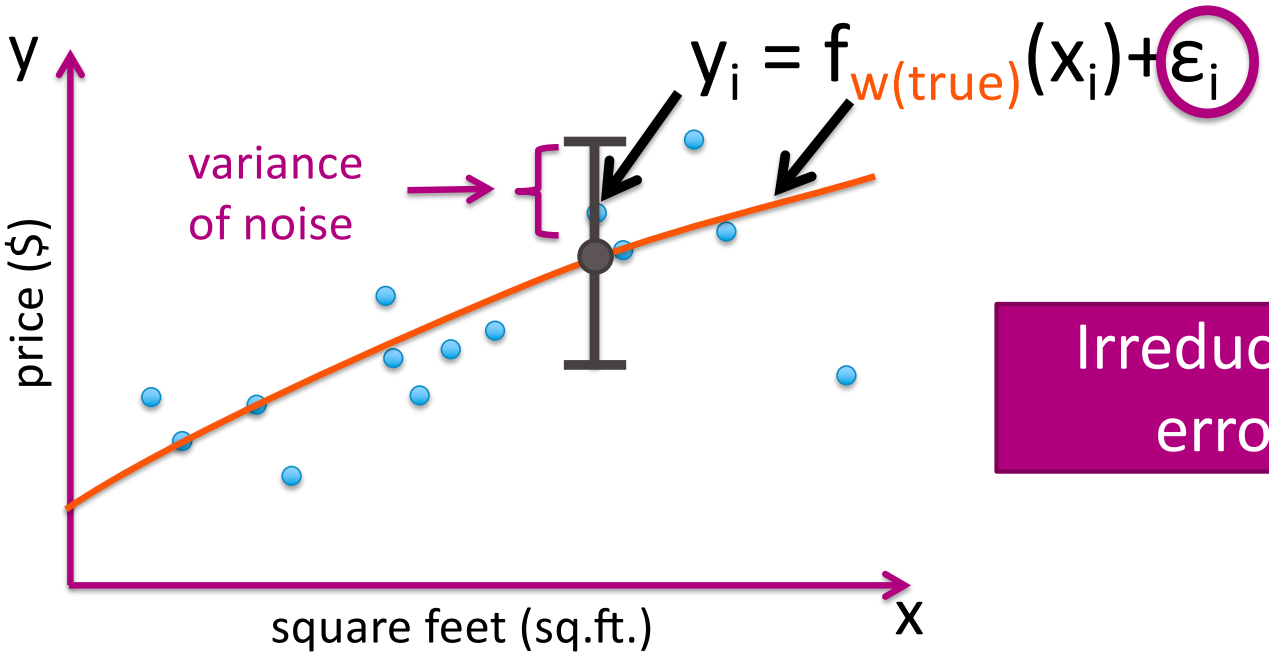
1. Noise
2. Bias
3. Variance

Data inherently noisy

Irreducible error = noise =

$E\left[(y - f_{w(true)}(x))^2\right]$

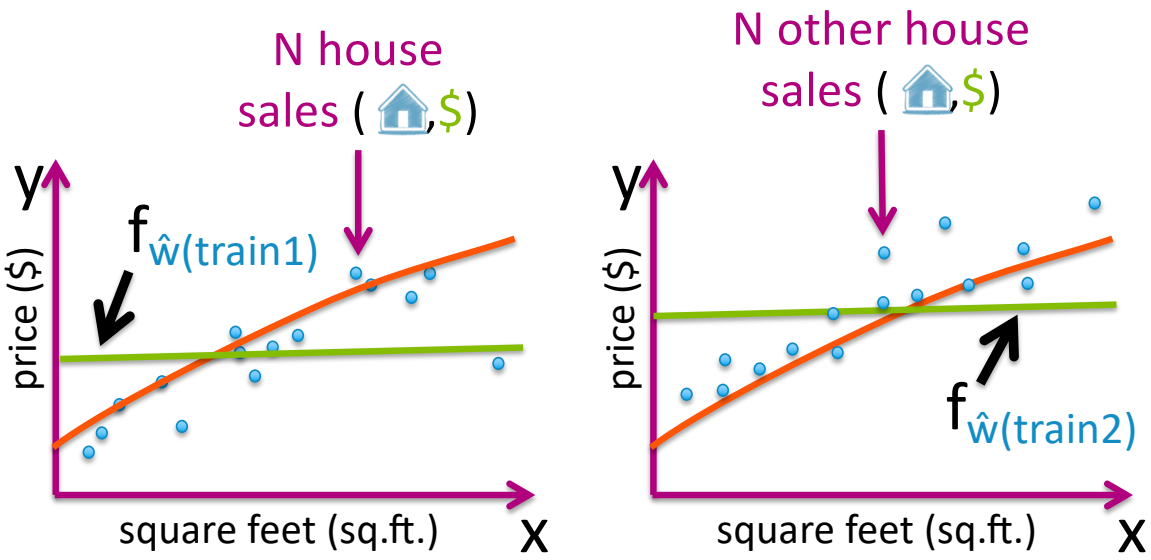
$f_{w(true)}(x) = E\left[y \mid x\right]$



Irreducible error

Bias contribution

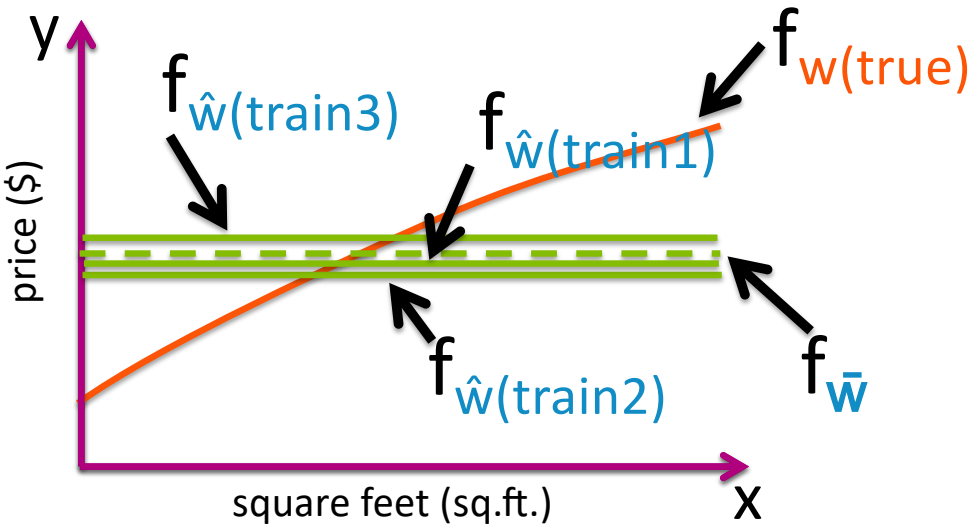
Assume we fit a constant function



Bias contribution

$$f_{\bar{w}}(x_t) = E_{train}\, f[_{\hat{w}(train)}(x_t)]$$

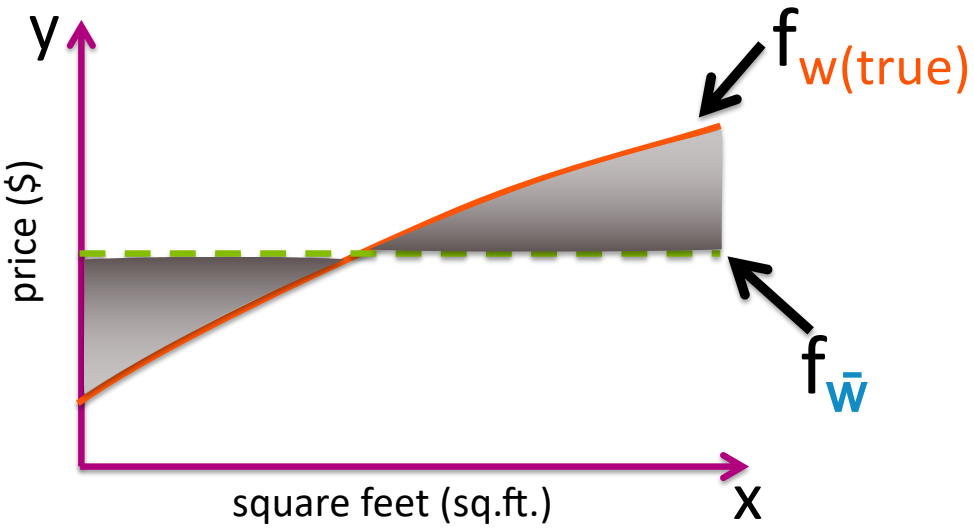
Over all possible size N training sets, what do I expect my fit to be?



Bias contribution

$\text{Bias}(x) = f_{w(\text{true})}(x) - f_{\bar{w}}(x)$

Is our approach flexible enough to capture $f_{w(\text{true})}$?
If not, error in predictions.

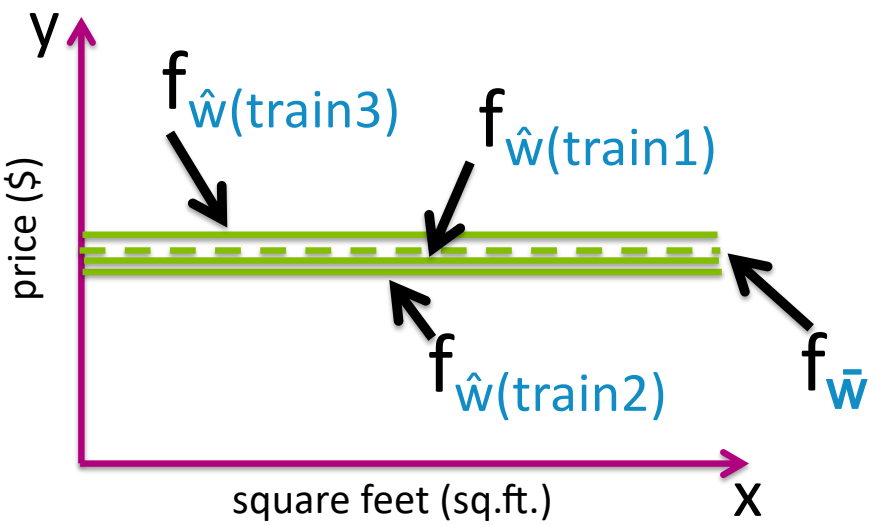


low complexity → high bias

Variance contribution

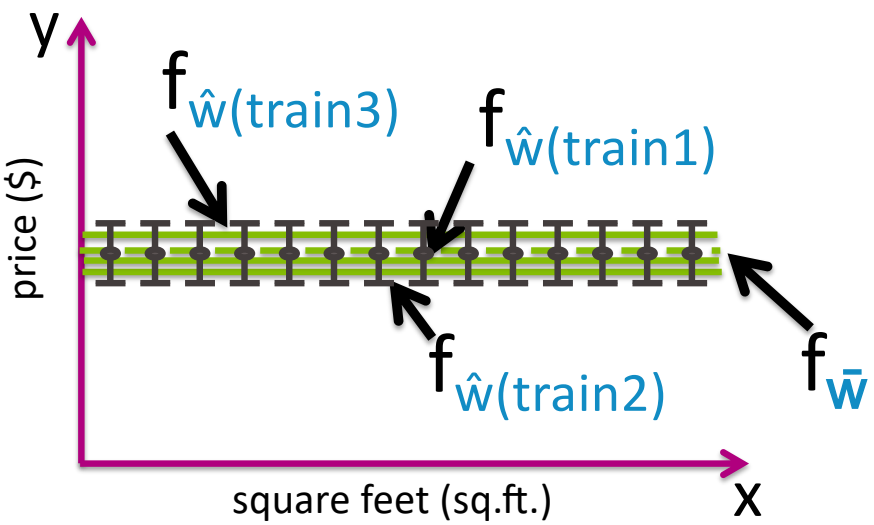
$$f_{\bar{w}}(x_t) = E_{train}\, f[_{\hat{w}(train)}(x_t)]$$

How much do specific fits vary from the expected fit?



Variance contribution

How much do specific fits vary from the expected fit?



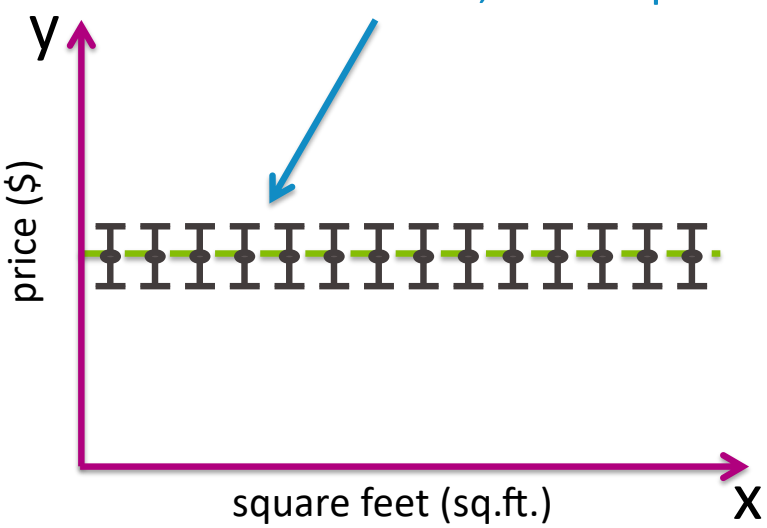
Variance contribution

$$\text{Variance}(x_t) = E_{\text{train}}\left[\left(f_{\hat{w}(\text{train})}(x_t) - f_{\bar{w}}(x_t)\right)^2\right]$$

$$f_{\bar{w}}(x_t) = E_{\text{train}}\left[f_{\hat{w}(\text{train})}(x_t)\right]$$

How much do specific fits vary from the expected fit?

Can specific fits vary widely?
If so, erratic predictions



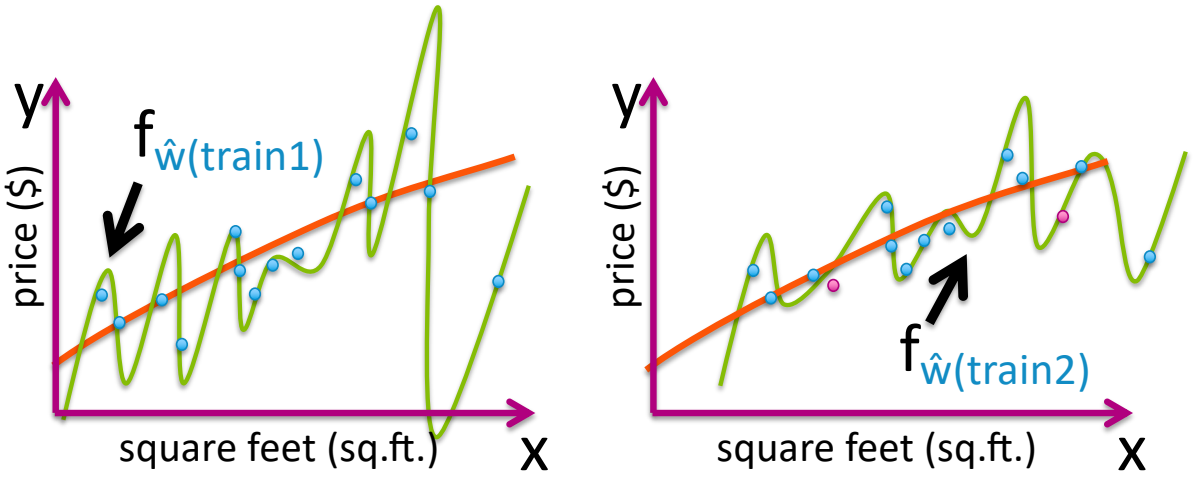
low complexity
→
low variance

Variance of high-complexity models

$$\text{Variance}(x_t) = E_{\text{train}}\left[\left(f_{\hat{w}(\text{train})}(x_t) - f_{\bar{w}}(x_t)\right)^2\right]$$

$$f_{\bar{w}}(x_t) = E_{\text{train}}\left[f_{\hat{w}(\text{train})}(x_t)\right]$$

Assume we fit a high-order polynomial

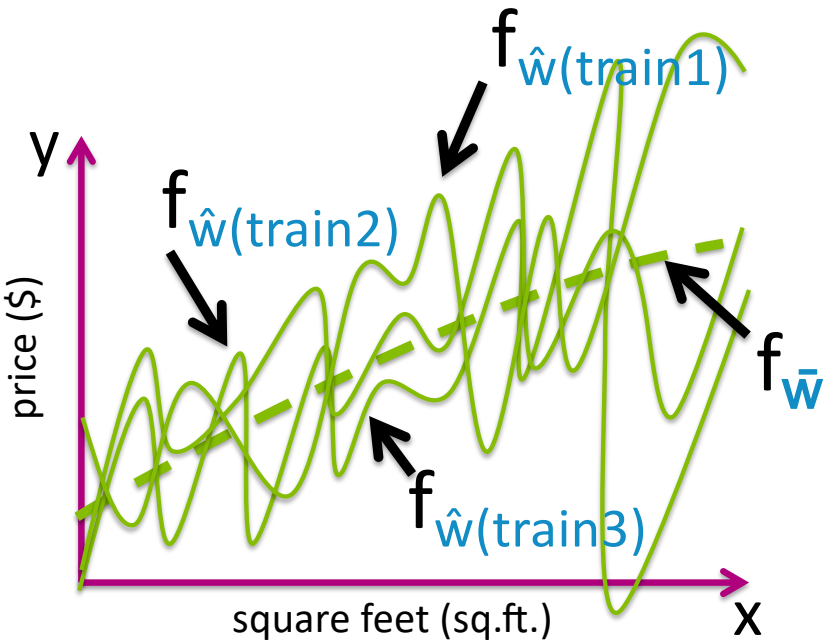


Variance of high-complexity models

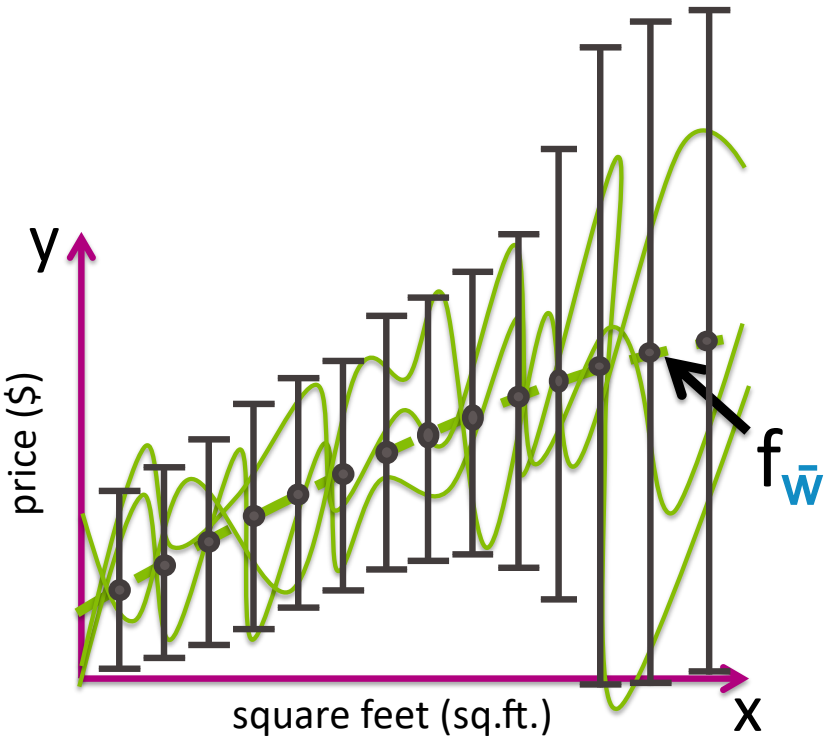
$$\text{Variance}(x_t) = E_{train}\left[\left(f_{\hat{w}(train)}(x_t) - f_{\bar{w}}(x_t)\right)^2\right]$$

$$f_{\bar{w}}(x_t) = E_{train}\left[f_{\hat{w}(train)}(x_t)\right]$$

Assume we fit a high-order polynomial



Variance of high-complexity models

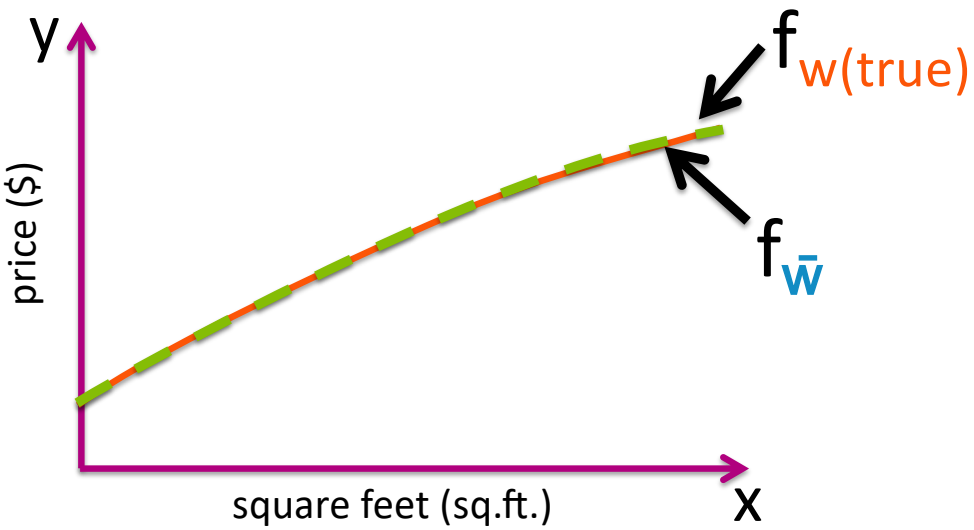


high complexity
→
high variance

Bias of high-complexity models

$$\text{Bias}(x) = f_{w(true)}(x) - f_{\bar{w}}(x)$$

$$f_{\bar{w}}(x_t) = E_{train}\, f[\hat{w}_{(train)}(x_t)]$$



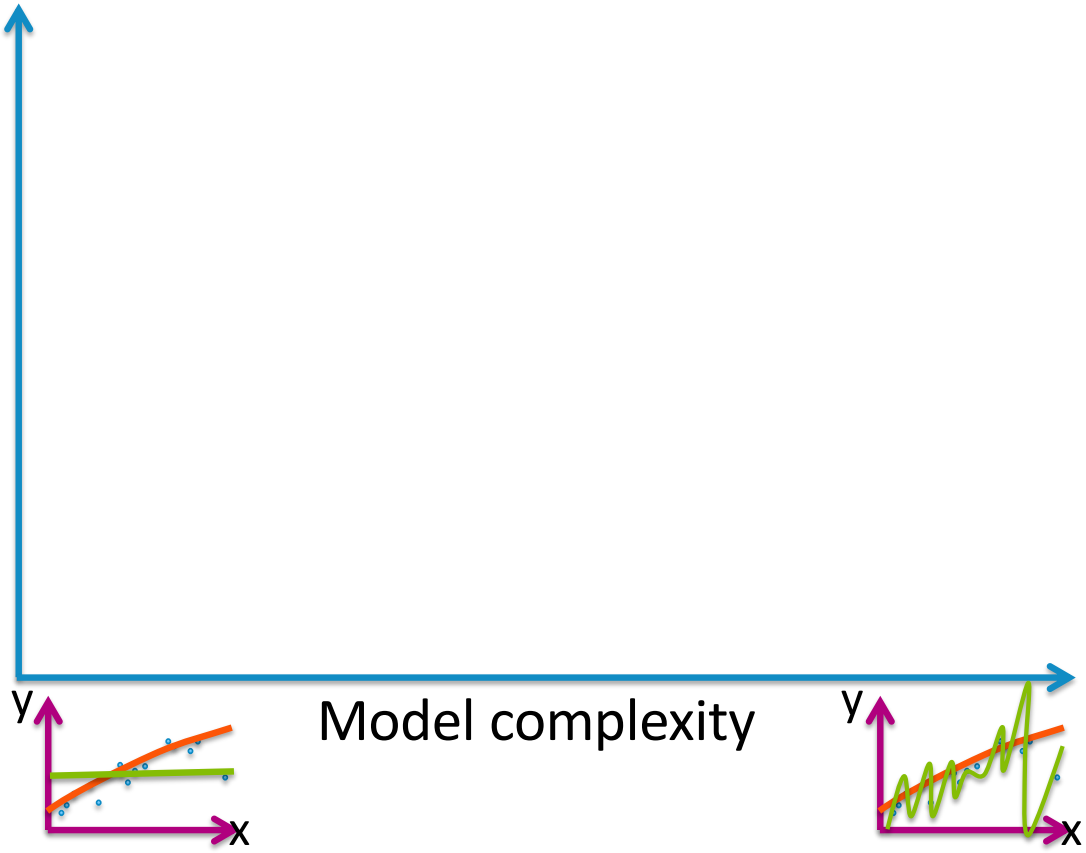
high complexity
→
low bias

Bias-variance tradeoff

$$\text{Bias}(x_t) = f_{w(\text{true})}(x_t) - f_{\bar{w}}(x_t)$$

$$\text{Variance}(x_t) = E_{\text{train}}\left[\left(f_{\hat{w}(\text{train})}(x_t) - f_{\bar{w}}(x_t)\right)^2\right]$$

$$f_{\bar{w}}(x_t) = E_{\text{train}}\, f[_{\hat{w}(\text{train})}(x_t)]$$



Error vs. amount of data **for fixed model complexity**

$$\text{Bias}(x_t) = f_{w(\text{true})}(x_t) - f_{\bar{w}}(x_t)$$

$$\text{Variance}(x_t) = E_{\text{train}}\left[\left(f_{\hat{w}(\text{train})}(x_t) - f_{\bar{w}}(x_t)\right)^2\right]$$

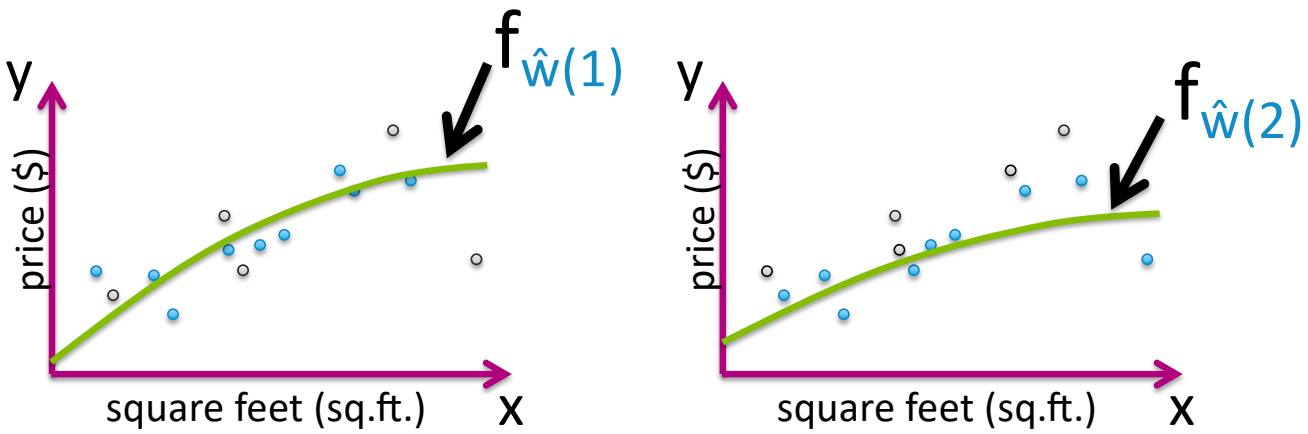
$$f_{\bar{w}}(x_t) = E_{\text{train}}\, f[_{\hat{w}(\text{train})}(x_t)]$$

More in depth on the
3 sources of errors…

Accounting for training set randomness

Training set was just a random sample of n houses sold

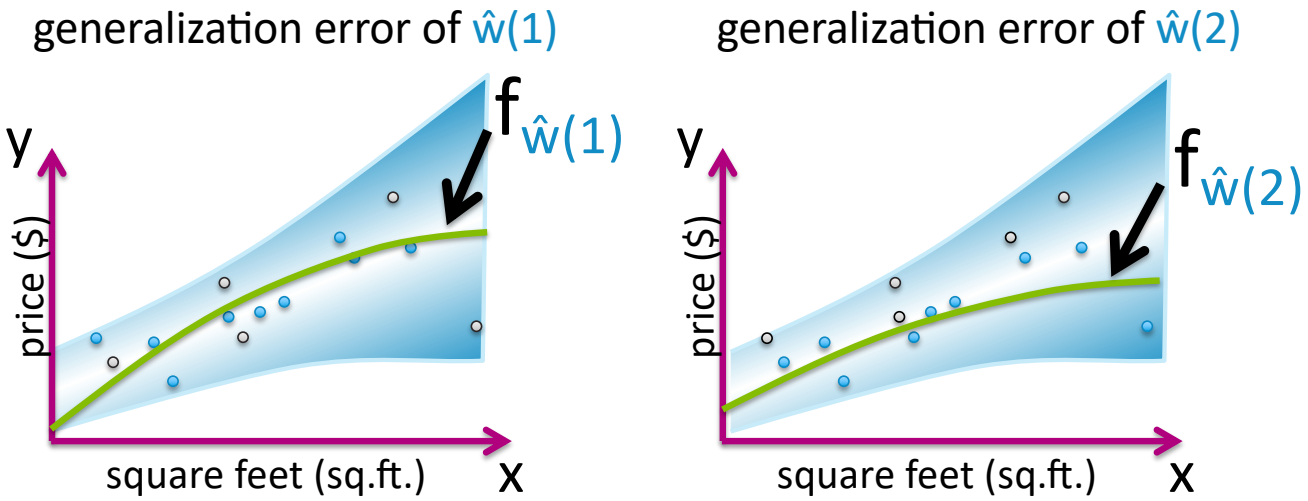
What if n other houses had been sold and recorded?



Accounting for training set randomness

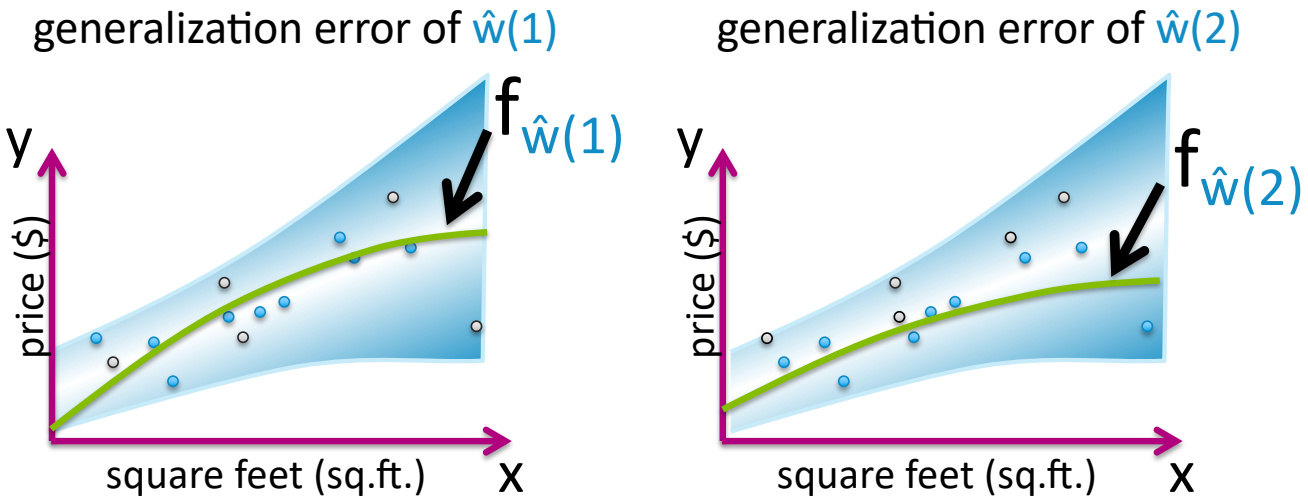
Training set was just a random sample of n houses sold

What if n other houses had been sold and recorded?



Accounting for training set randomness

Ideally, want performance averaged over all possible training sets of size n

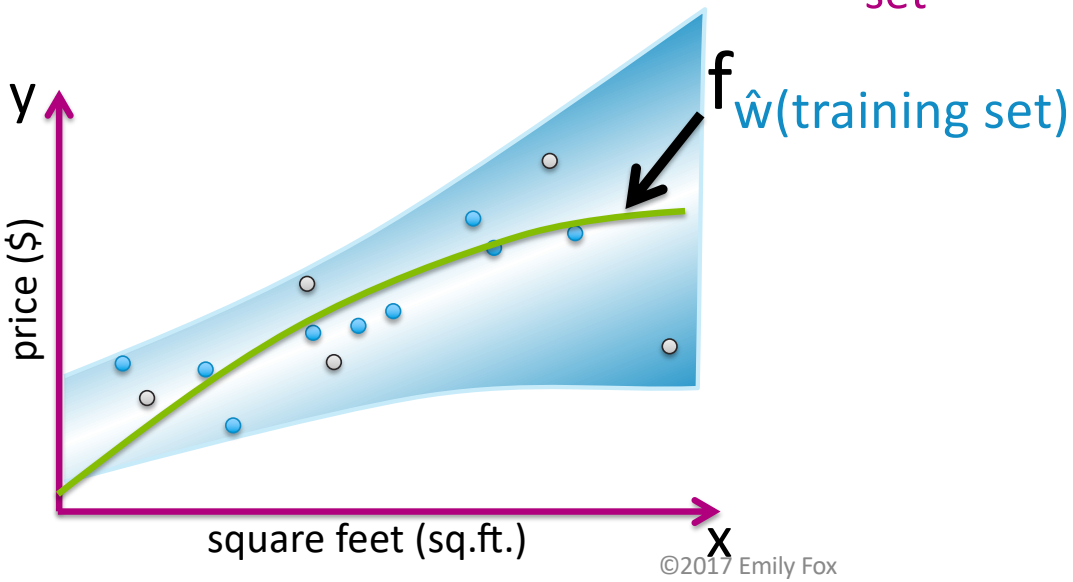


Expected prediction error

$E_{\text{training set}}$[generalization error of $\hat{w}$(training set)]

averaging over all training sets (weighted by how likely each is)

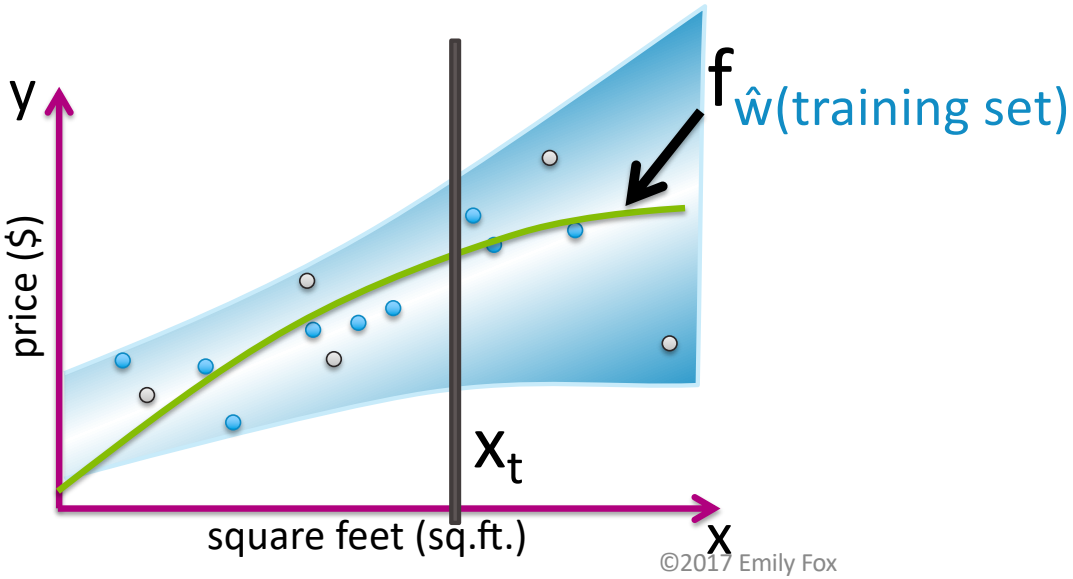
parameters fit on a specific training set



Prediction error at target input

Start by considering:

1. Loss at target x_t (e.g. 2640 sq.ft.)
2. Squared error loss $L(y,f_{\hat{w}}(x)) = (y-f_{\hat{w}}(x))^2$



Sum of 3 sources of error

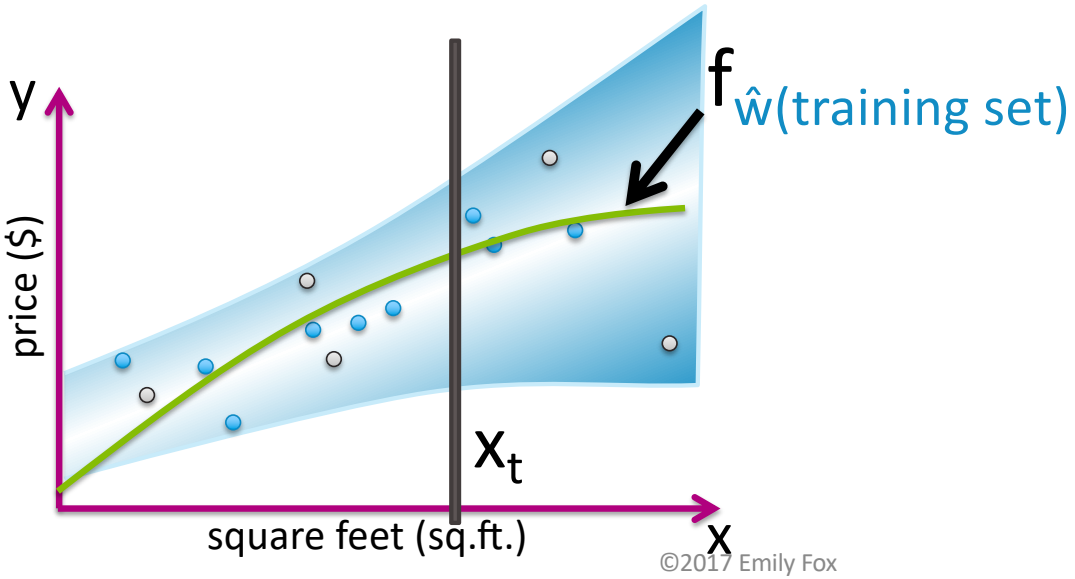
$$\text{Bias}(x_t) = f_{w(true)}(x_t) - f_{\bar{w}}(x_t)$$

$$\text{Variance}(x_t) = E_{train}\left[\left(f_{\hat{w}(train)}(x_t) - f_{\bar{w}}(x_t)\right)^2\right]$$

$$f_{\bar{w}}(x_t) = E_{train}\, f[_{\hat{w}(train)}(x_t)]$$

Average prediction error at x_t

$$= \sigma^2 + \left[\text{bias}(f_{\hat{w}}(x_t))\right]^2 + \text{var}(f_{\hat{w}}(x_t))$$



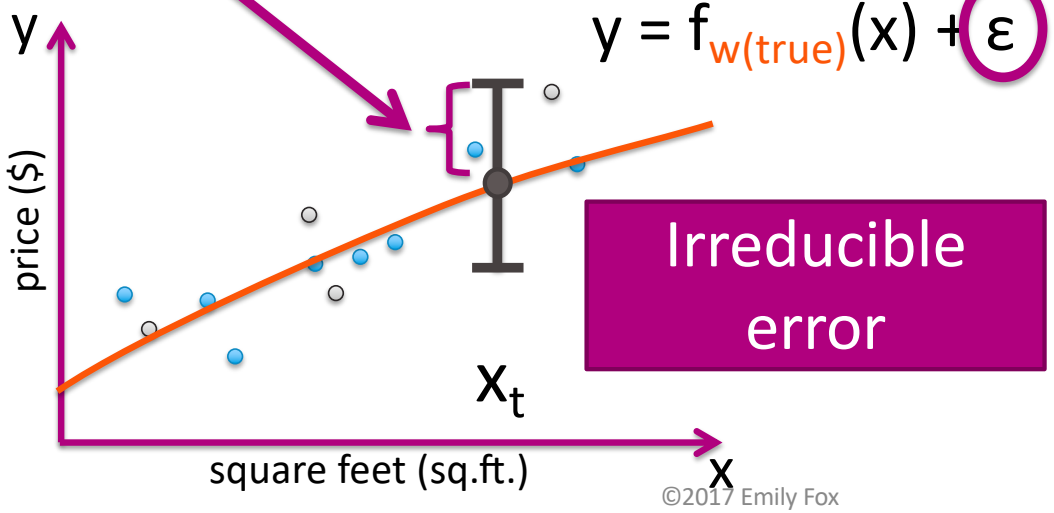
Error variance of the model

Average prediction error at x_t

$$= \sigma^2 + [\text{bias}(f_{\hat{w}}(x_t))]^2 + \text{var}(f_{\hat{w}}(x_t))$$

σ^2 ="variance"

$$y = f_{w(true)}(x) + \varepsilon$$



Bias of function estimator

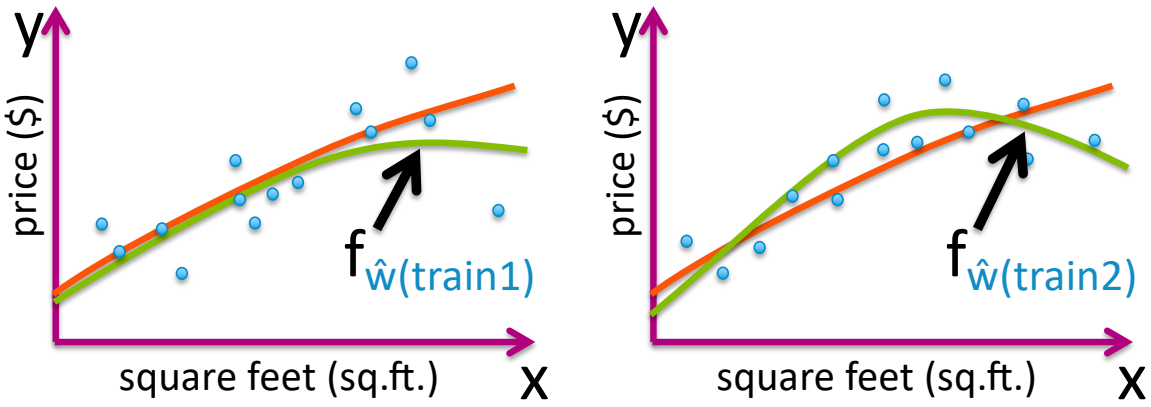
$$\text{Bias}(x_t) = f_{w(\text{true})}(x_t) - f_{\bar{w}}(x_t)$$

$$\text{Variance}(x_t) = E_{\text{train}}\left[\left(f_{\hat{w}(\text{train})}(x_t) - f_{\bar{w}}(x_t)\right)^2\right]$$

$$f_{\bar{w}}(x_t) = E_{\text{train}}\, f[_{\hat{w}(\text{train})}(x_t)]$$

Average prediction error at x_t

$$= \sigma^2 + [\text{bias}(f_{\hat{w}}(x_t))]^2 + \text{var}(f_{\hat{w}}(x_t))$$



Bias of function estimator

$$\text{Bias}(x_t) = f_{w(true)}(x_t) - f_{\bar{w}}(x_t)$$

$$\text{Variance}(x_t) = E_{train}\left[\left(f_{\hat{w}(train)}(x_t) - f_{\bar{w}}(x_t)\right)^2\right]$$

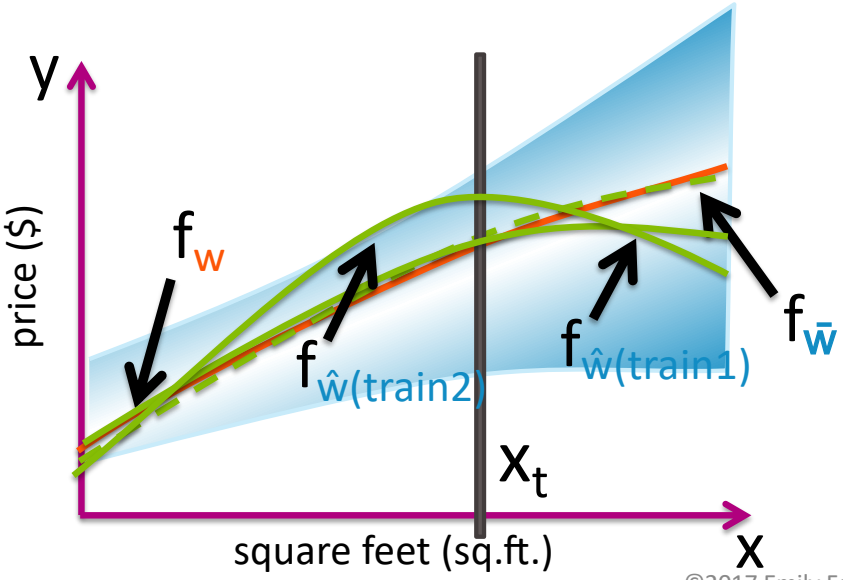
$$f_{\bar{w}}(x_t) = E_{train}\, f[_{\hat{w}(train)}(x_t)]$$

Average estimated function = $f_{\bar{w}}(x)$

True function = $f_w(x)$

$$E_{train}[f_{\hat{w}(train)}(x)]$$

over all training sets of size N



Bias of function estimator

$$\text{Bias}(x_t) = f_{w(\text{true})}(x_t) - f_{\bar{w}}(x_t)$$

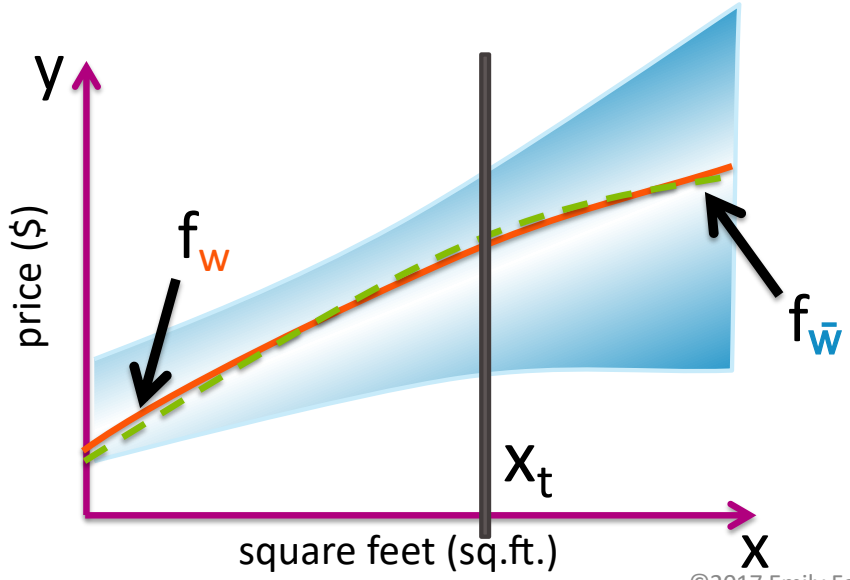
$$\text{Variance}(x_t) = E_{\text{train}}\left[\left(f_{\hat{w}(\text{train})}(x_t) - f_{\bar{w}}(x_t)\right)^2\right]$$

$$f_{\bar{w}}(x_t) = E_{\text{train}}\, f[_{\hat{w}(\text{train})}(x_t)]$$

Average estimated function = $f_{\bar{w}}(x)$

True function = $f_w(x)$

$$\text{bias}(f_{\hat{w}}(x_t)) = f_w(x_t) - f_{\bar{w}}(x_t)$$



Bias of function estimator

Average prediction error at x_t

$$= \sigma^2 + [\text{bias}(f_{\hat{w}}(x_t))]^2 + \text{var}(f_{\hat{w}}(x_t))$$

$$\text{Bias}(x_t) = f_{w(\text{true})}(x_t) - f_{\bar{w}}(x_t)$$

$$\text{Variance}(x_t) = E_{\text{train}}[(f_{\hat{w}(\text{train})}(x_t) - f_{\bar{w}}(x_t))^2]$$

$$f_{\bar{w}}(x_t) = E_{\text{train}}\, f[_{\hat{w}(\text{train})}(x_t)]$$

Variance of function estimator

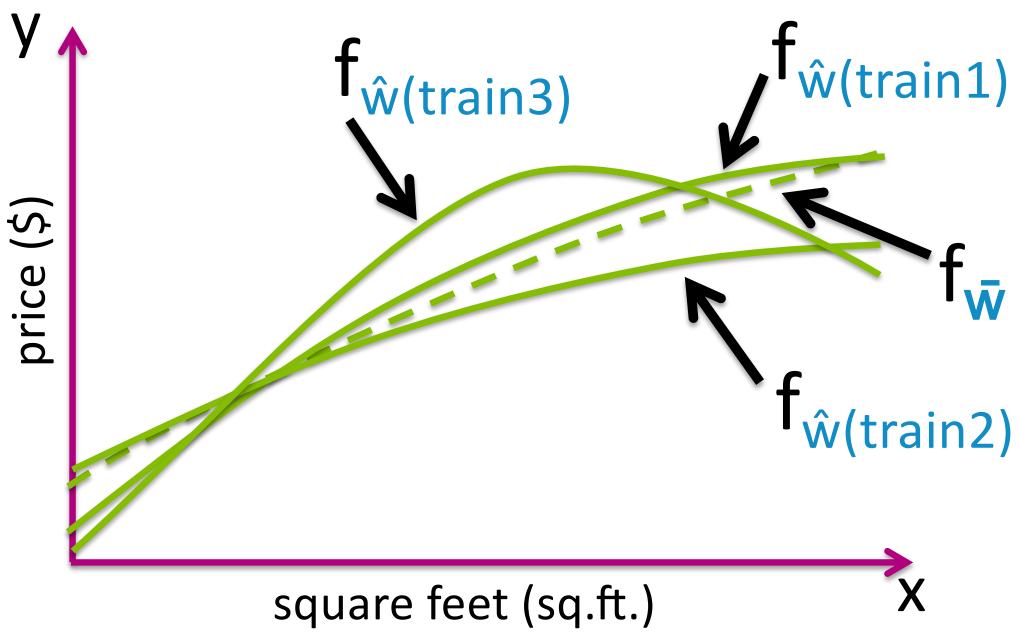
$$\text{Bias}(x_t) = f_{w(true)}(x_t) - f_{\bar{w}}(x_t)$$

$$\text{Variance}(x_t) = E_{train}\left[\left(f_{\hat{w}(train)}(x_t) - f_{\bar{w}}(x_t)\right)^2\right]$$

$$f_{\bar{w}}(x_t) = E_{train}\, f[_{\hat{w}(train)}(x_t)]$$

Average prediction error at x_t

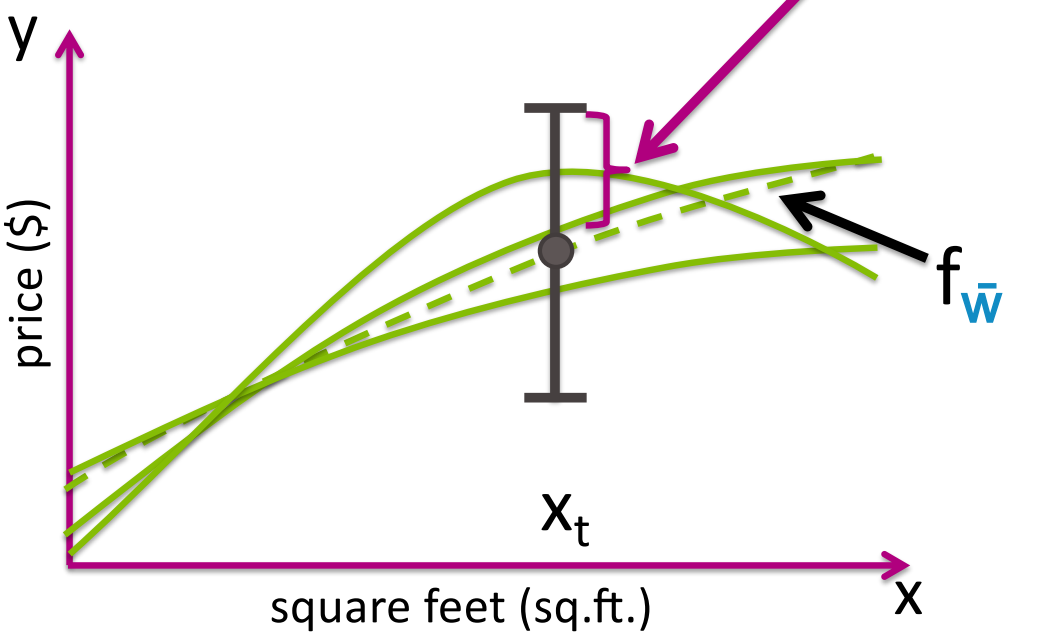
$$= \sigma^2 + [\text{bias}(f_{\hat{w}}(x_t))]^2 + \text{var}(f_{\hat{w}}(x_t))$$



Variance of function estimator

Average prediction error at x_t

$$= \sigma^2 + [\text{bias}(f_{\hat{w}}(x_t))]^2 + \text{var}(f_{\hat{w}}(x_t))$$



Variance of function estimator

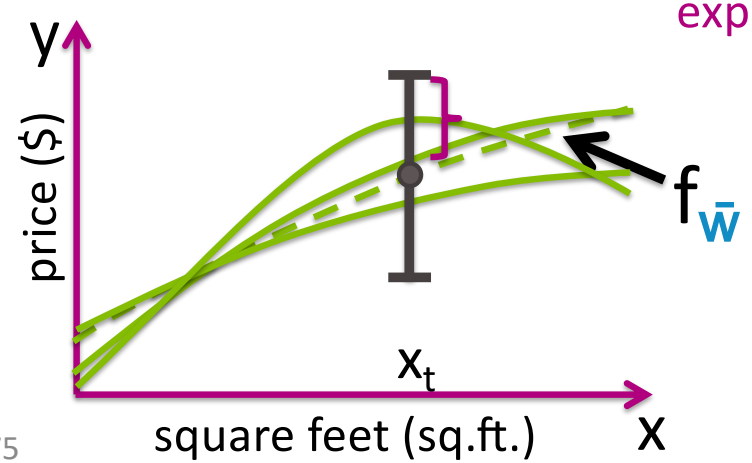
fit on a specific training dataset

what I expect to learn over all training sets

$$\text{var}(f_{\hat{w}}(x_t)) = E_{\text{train}}[(f_{\hat{w}(\text{train})}(x_t) - f_{\bar{w}}(x_t))^2]$$

over all training sets of size N

deviation of specific fit from expected fit at x_t



Why 3 sources of error?
A formal derivation

Deriving expected prediction error

Expected prediction error

$= E_{train}$ [generalization error of $\hat{w}(train)$]

$= E_{train}\left[E_{x,y}\left[L(y, f_{\hat{w}(train)}(x))\right]\right]$

1. Look at specific x_t
2. Consider $L(y, f_{\hat{w}}(x)) = (y - f_{\hat{w}}(x))^2$

Expected prediction error at x_t

$= E_{train,\, y_t}\left[(y_t - f_{\hat{w}(train)}(x_t))^2\right]$

Deriving expected prediction error

Expected prediction error at x_t

$$= E_{train, y_t}\left[(y_t - f_{\hat{w}(train)}(x_t))^2\right]$$

$$= E_{train, y_t}\left[((y_t - f_{w(true)}(x_t)) + (f_{w(true)}(x_t) - f_{\hat{w}(train)}(x_t)))^2\right]$$

Equating MSE with bias and variance

$$\text{MSE}\left[f_{\hat{w}(\text{train})}(x_t)\right]$$

$$= E_{\text{train}}\left[\left(f_{w(\text{true})}(x_t) - f_{\hat{w}(\text{train})}(x_t)\right)^2\right]$$

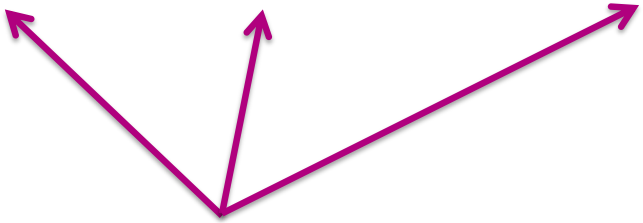
$$= E_{\text{train}}\left[\left(\left(f_{w(\text{true})}(x_t) - f_{\bar{w}}(x_t)\right) + \left(f_{\bar{w}}(x_t) - f_{\hat{w}(\text{train})}(x_t)\right)\right)^2\right]$$

Putting it all together

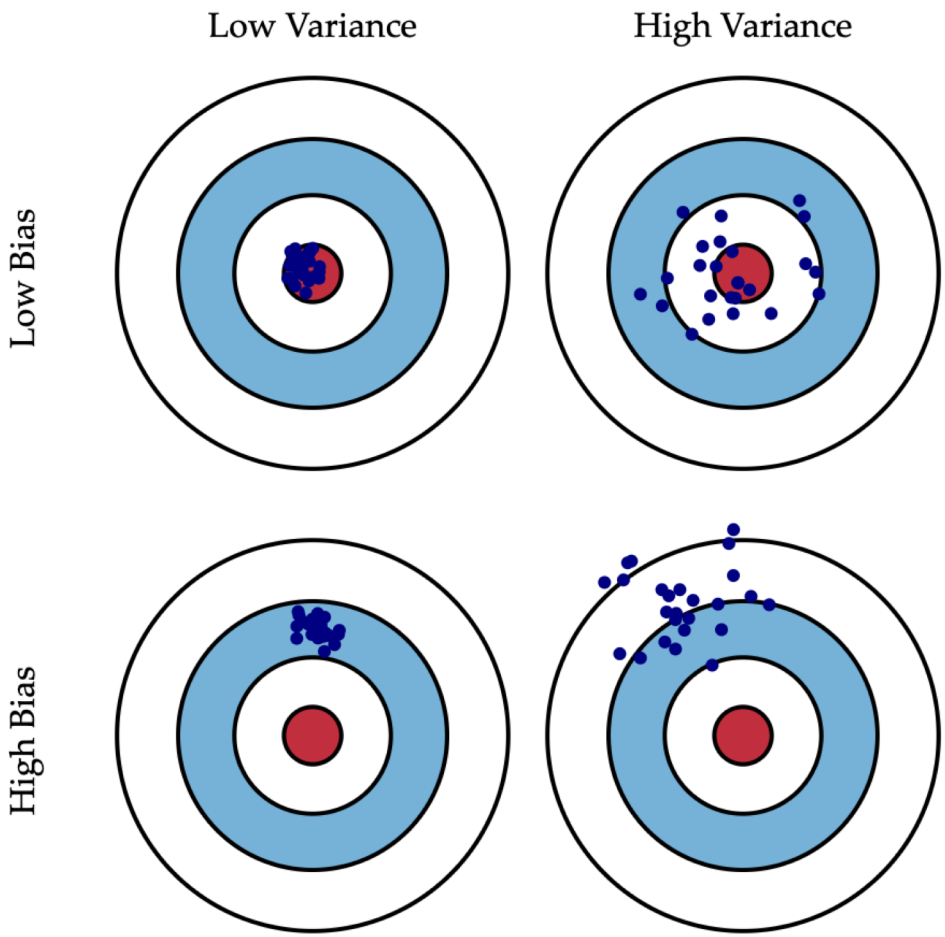
Expected prediction error at x_t

$$= \sigma^2 + \text{MSE}\left[f_{\hat{w}}(x_t)\right]$$

$$= \sigma^2 + \left[\text{bias}(f_{\hat{w}}(x_t))\right]^2 + \text{var}(f_{\hat{w}}(x_t))$$



3 sources of error



Courtesy of Scott Fortmann-Roe

Summary of assessing performance

What you can do now…

- Describe what a loss function is and give examples
- Contrast training, generalization, and test error
- Compute training and test error given a loss function
- Discuss issue of assessing performance on training set
- Describe tradeoffs in forming training/test splits
- Understand the 3 sources of avg. prediction error
  - Irreducible error, bias, and variance