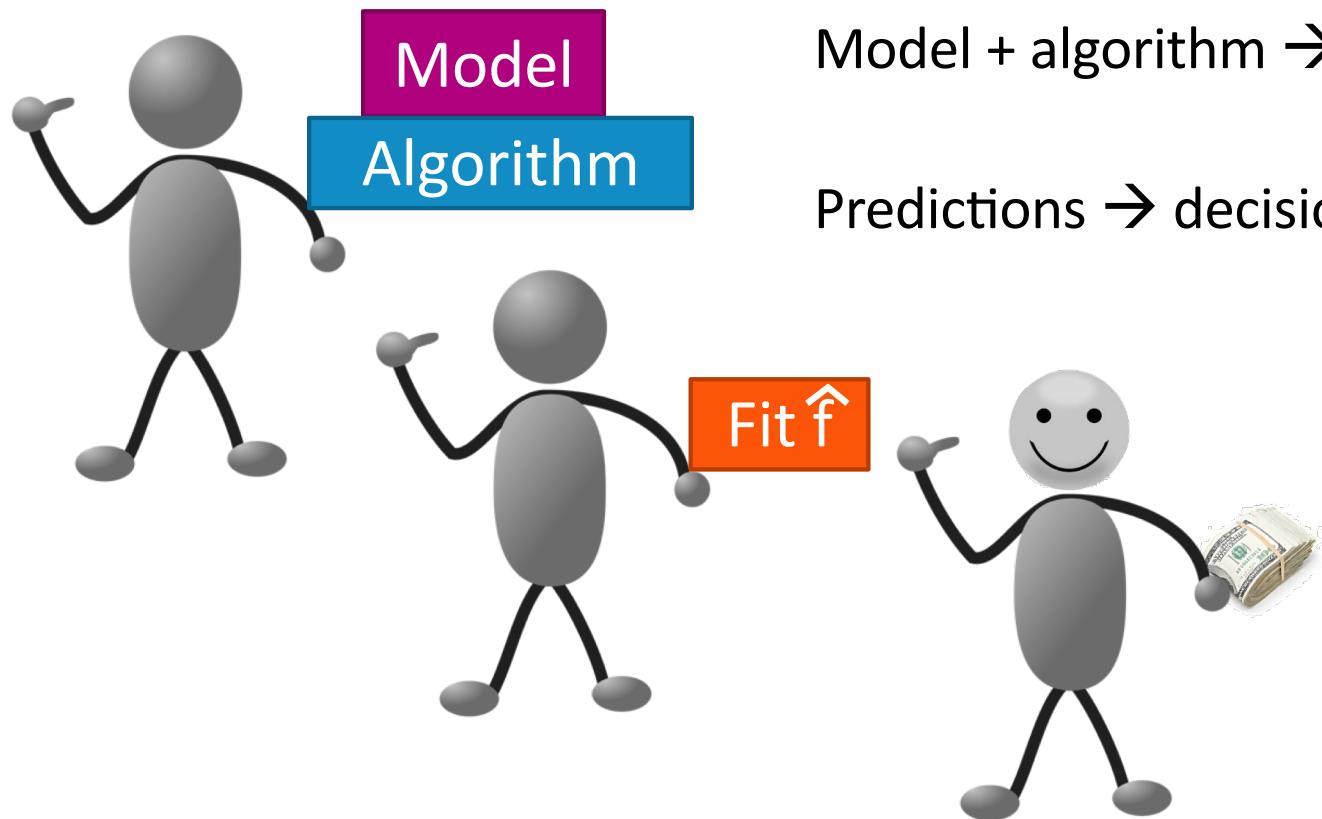


# Assessing Performance

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

April 8, 2019

# How good are my predictions?



Model + algorithm → fitted function

Predictions → decisions → outcome

# Measuring loss

Loss function:

$$L(y, \hat{f}_w(x))$$

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

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

Examples:

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

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

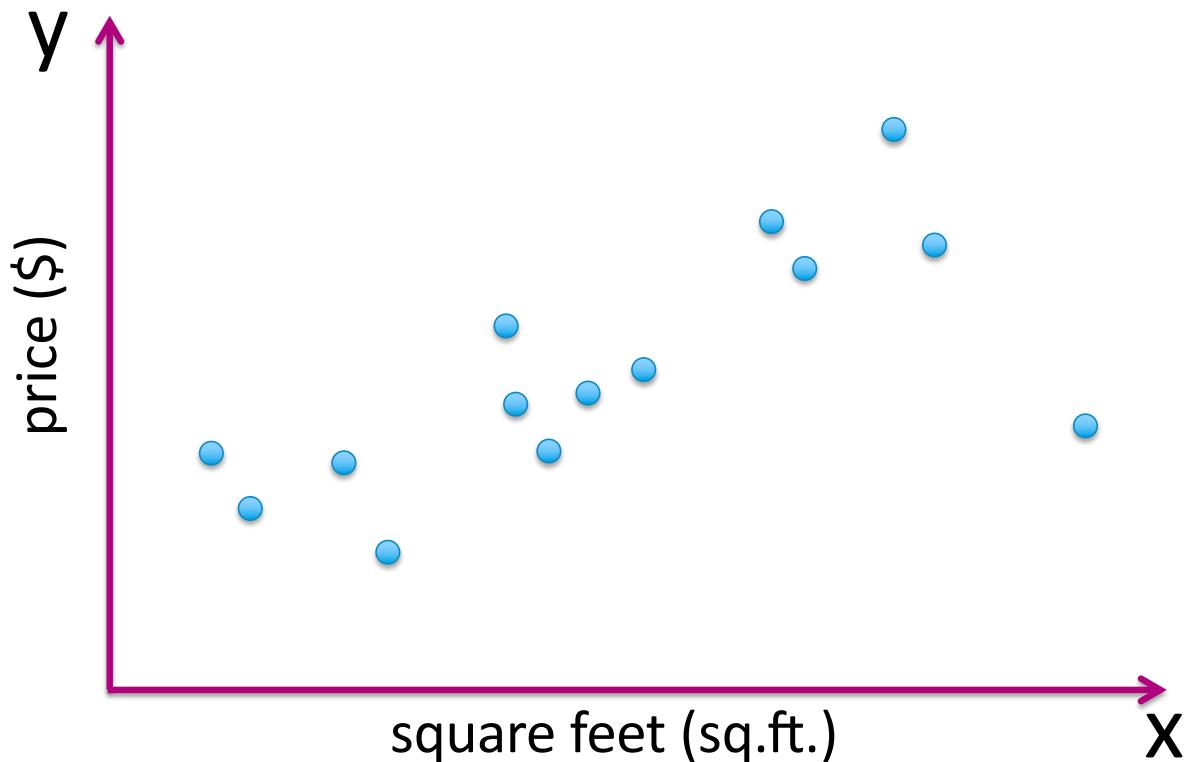
“Remember that all models are wrong; the practical question is how wrong do they have to be to not be useful.” George Box, 1987.

# *Assessing the loss*

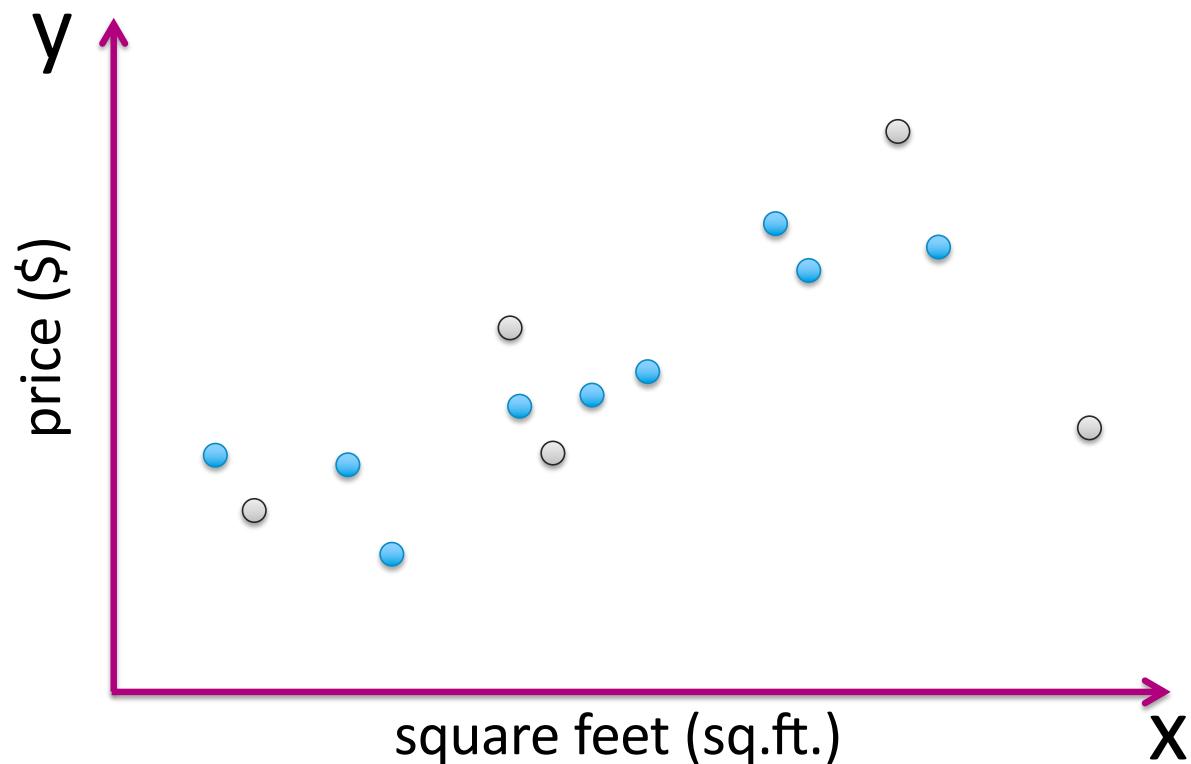
# Assessing the loss

## Part 1: Training error

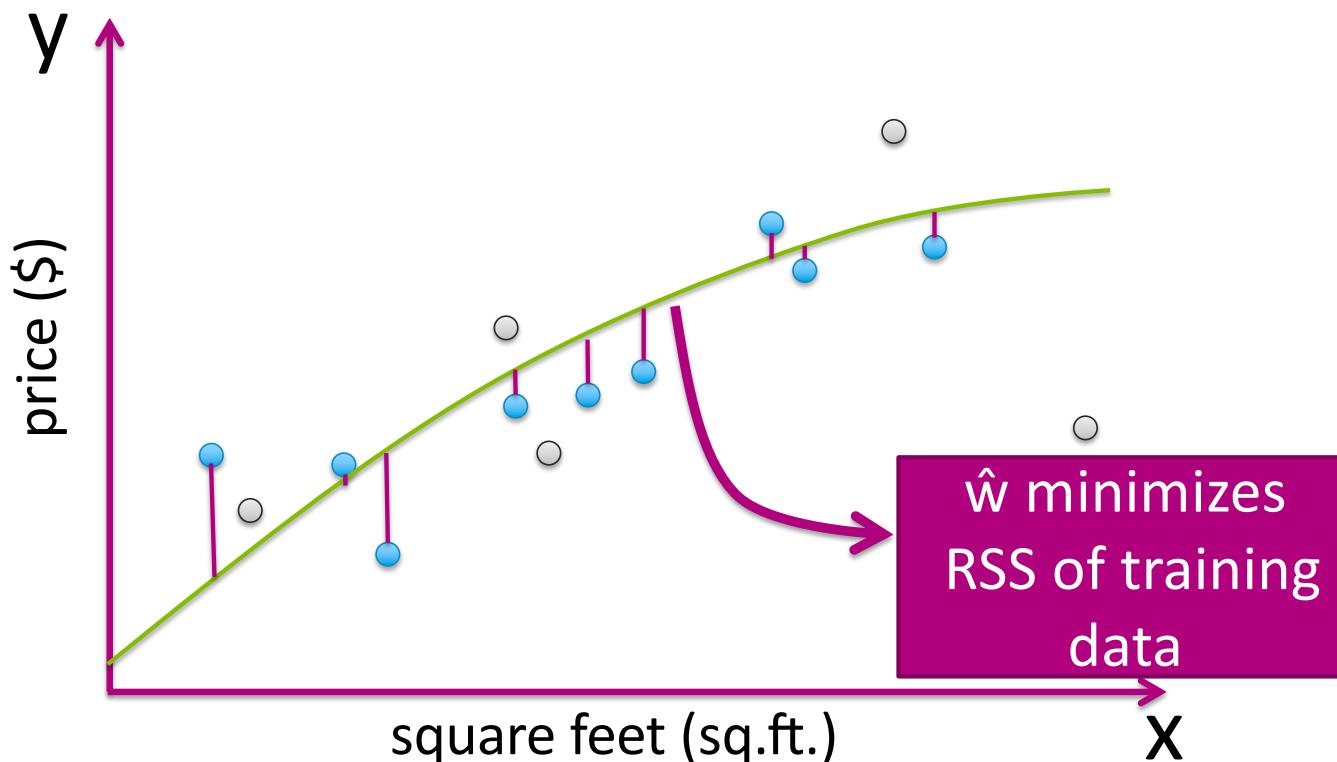
# Start with a data set



# 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,...

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

= avg. loss on houses in **training set**

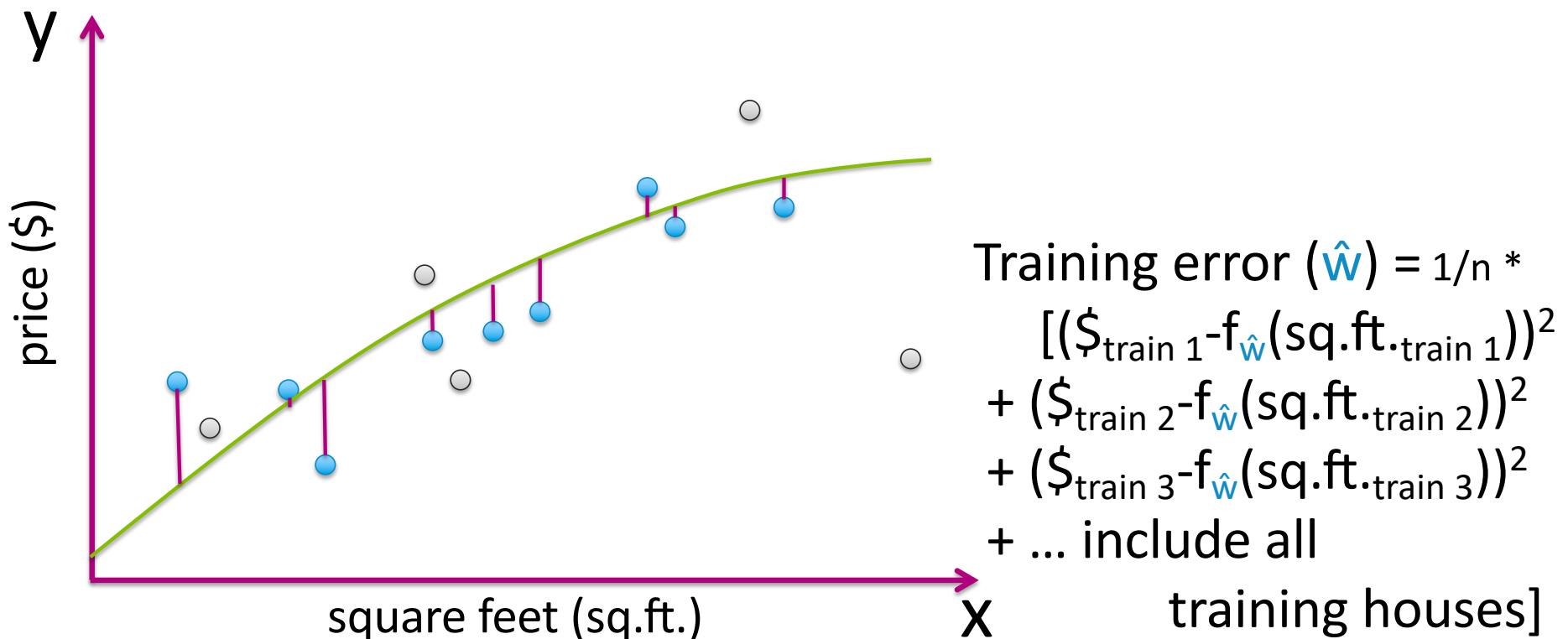
$$= \frac{1}{n} \sum_{i=1}^n L(y_i, f_{\hat{w}}(x_i))$$



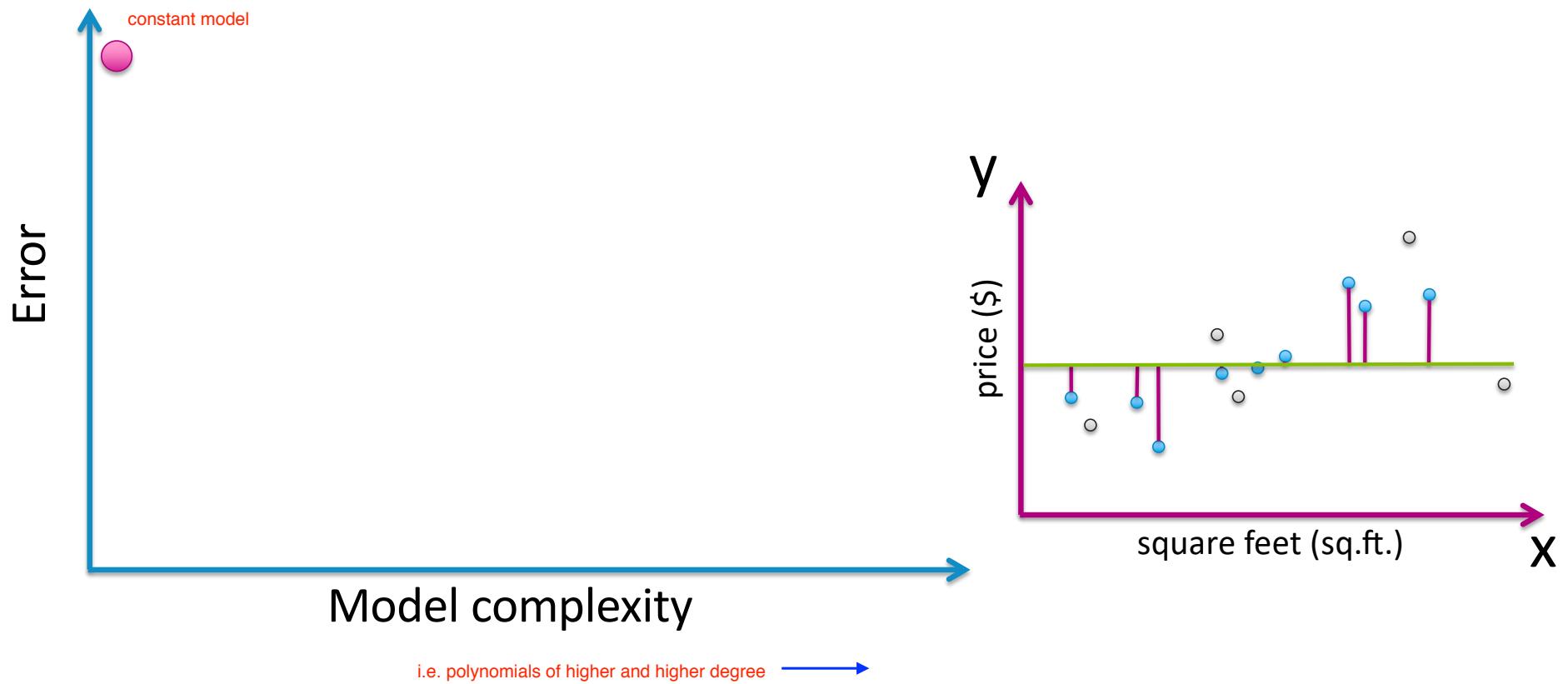
fit using training data

Example:

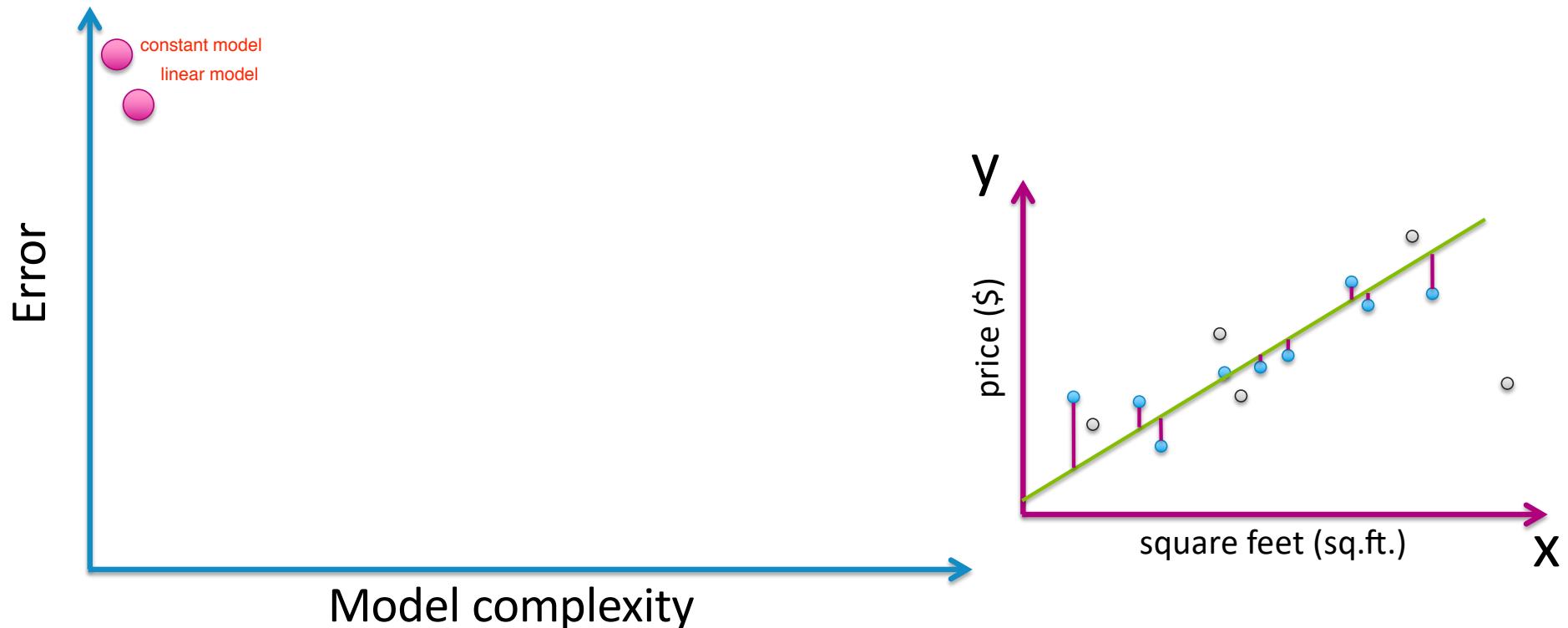
Use squared error loss  $(y - f_{\hat{w}}(x))^2$



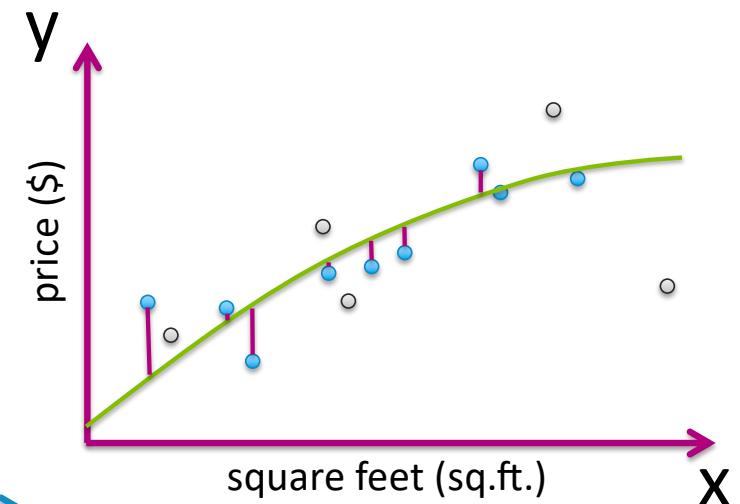
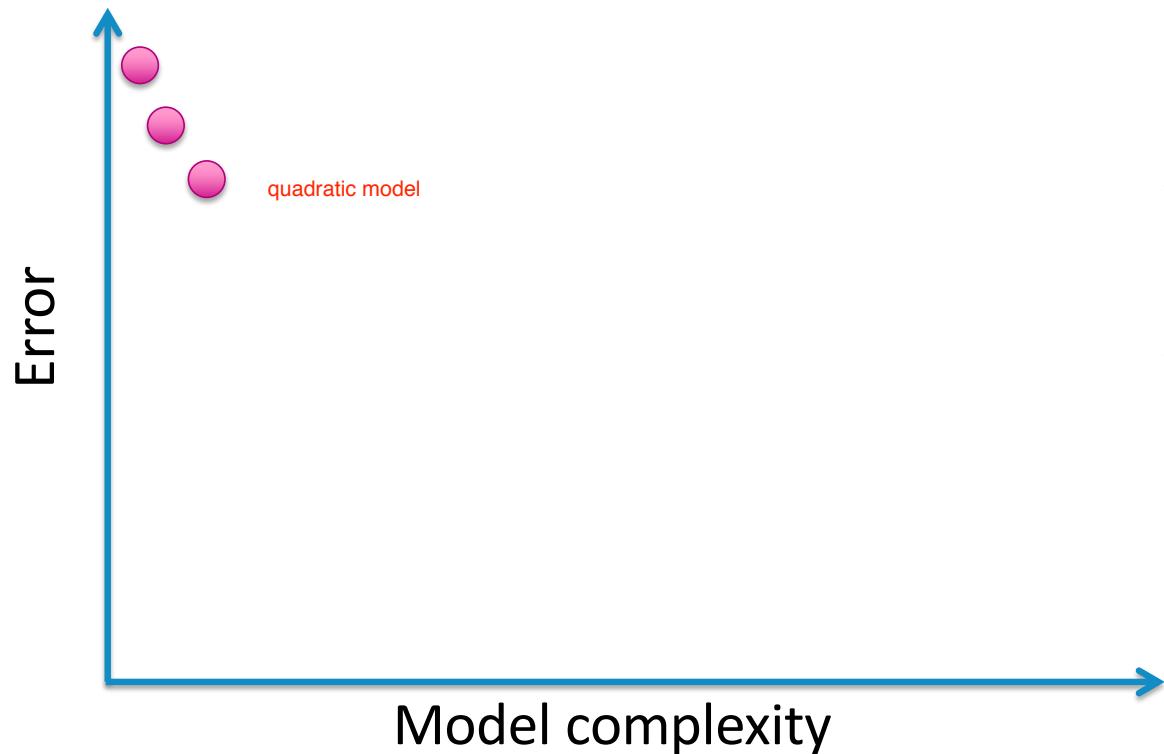
# Training error vs. model complexity



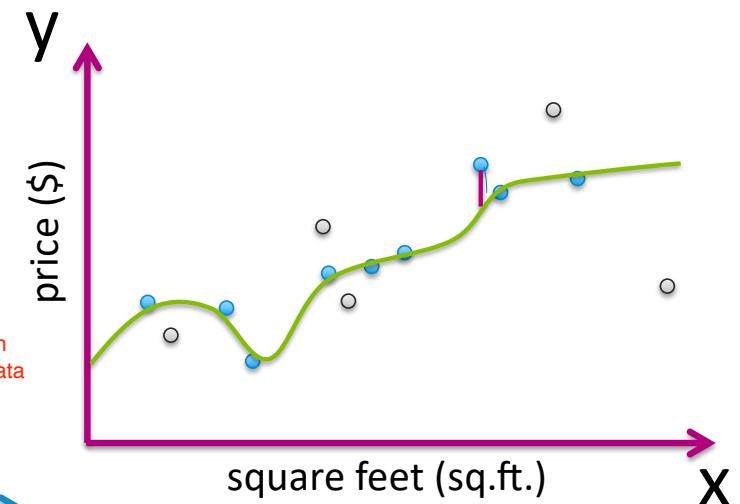
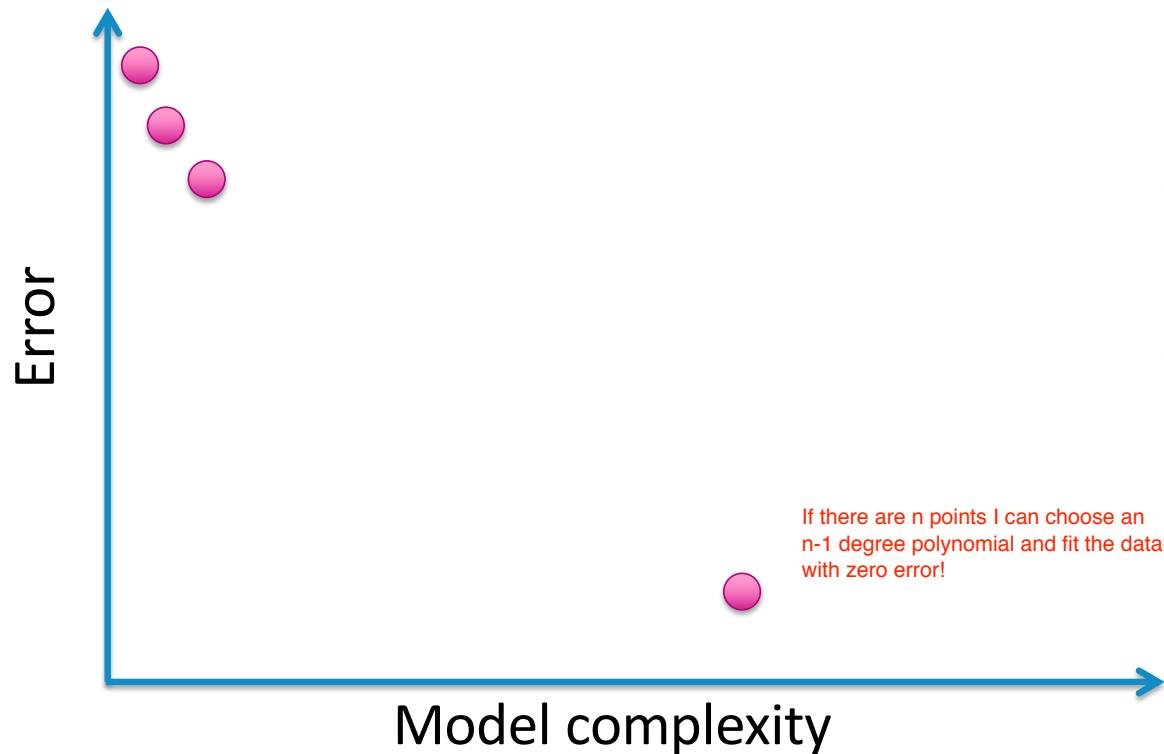
# Training error vs. model complexity



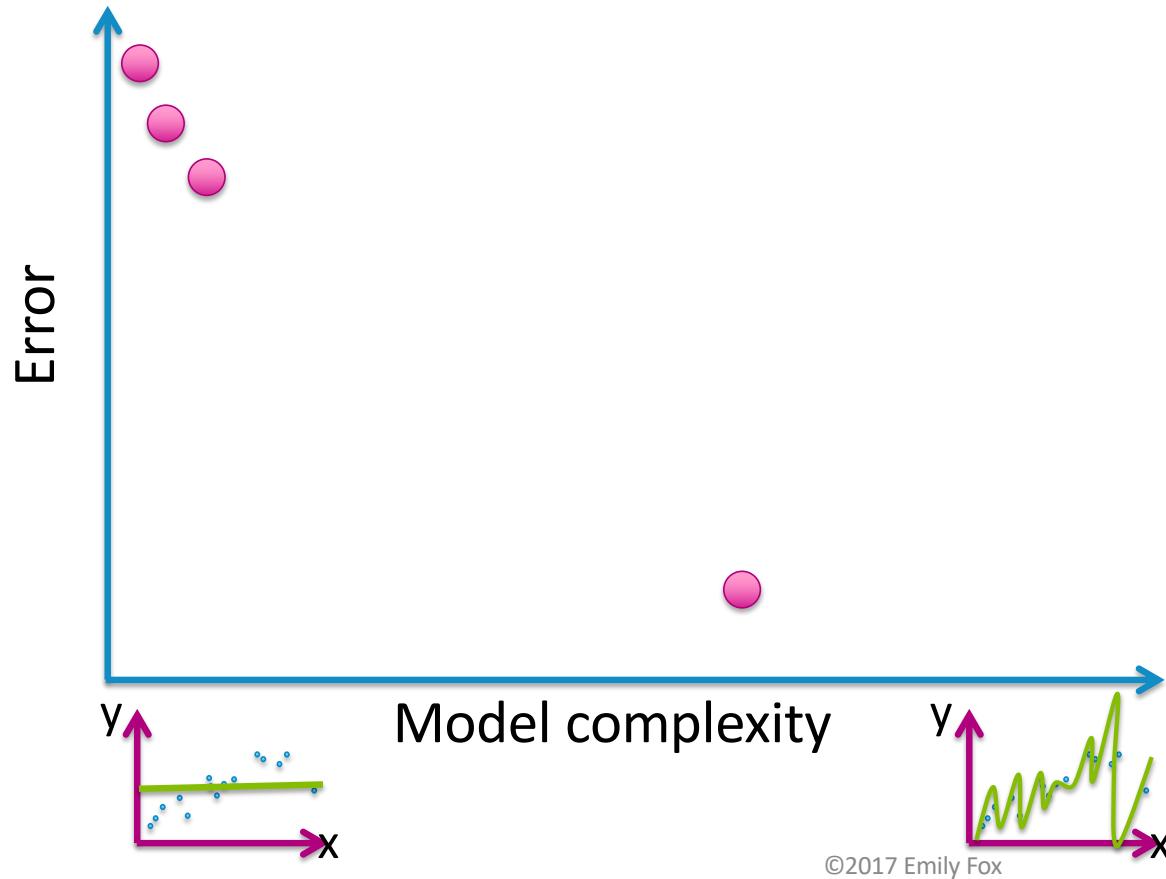
# Training error vs. model complexity



# Training error vs. model complexity

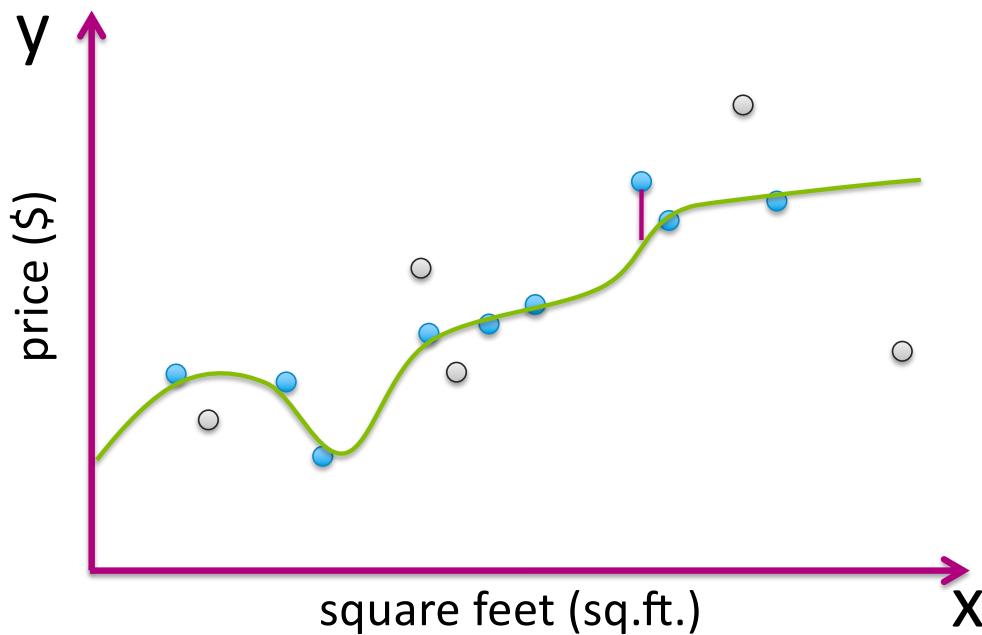


# Training error vs. model complexity



Is training error a good measure of predictive performance?

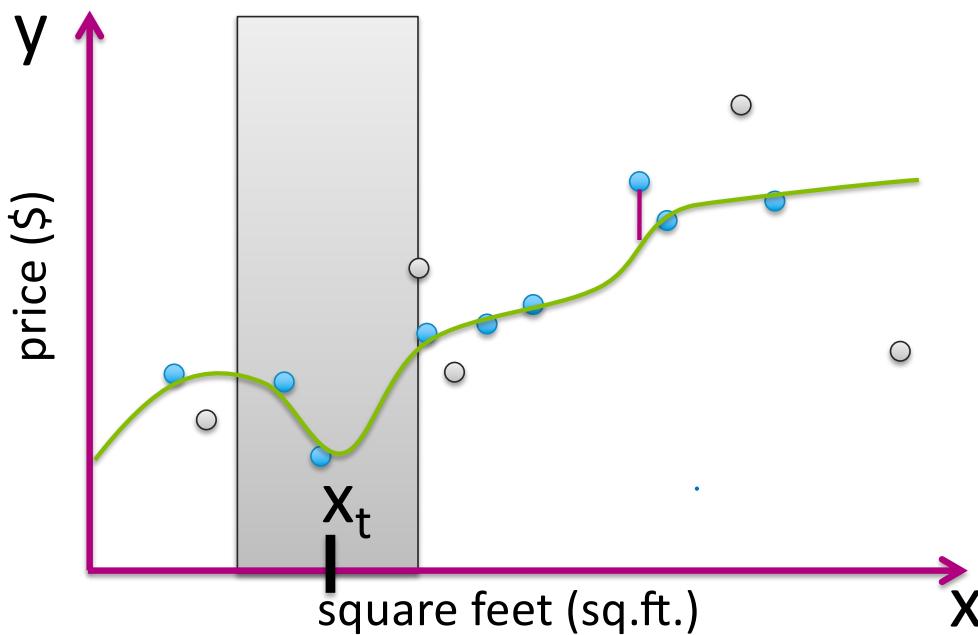
How do we expect to perform on a new house?



# Is training error a good measure of predictive performance?

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

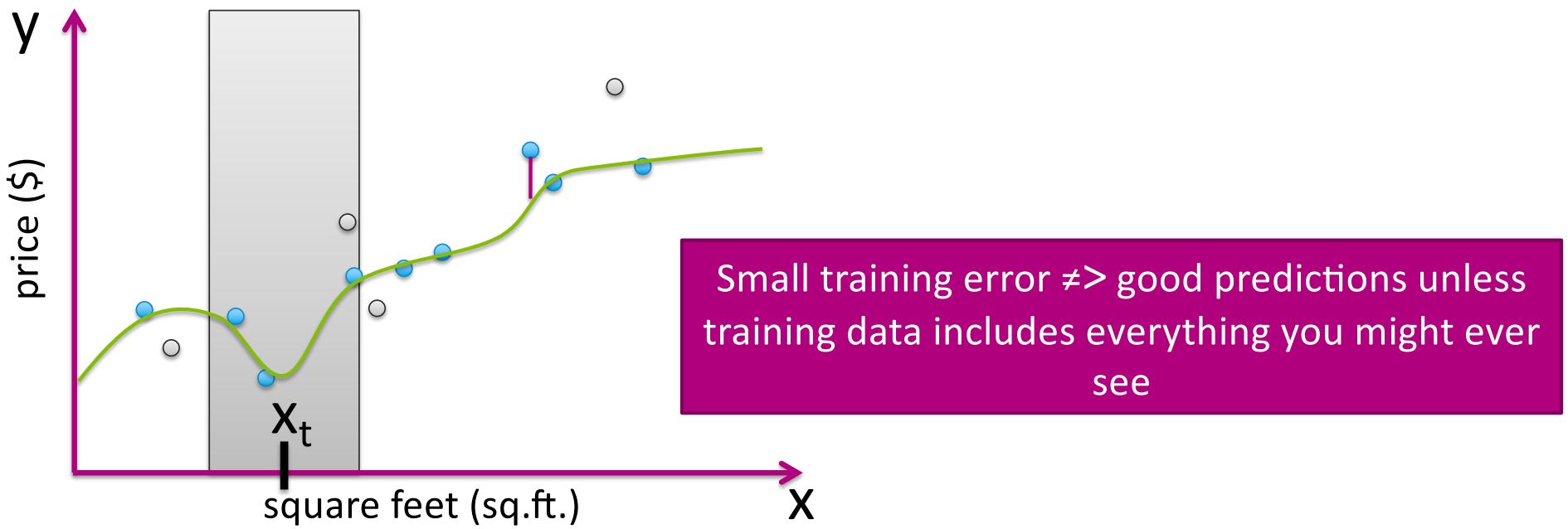
NO! We assumed that price was completely explained by differences in square feet! Instead, there are likely many other parameters as well as simple noise that are influencing changes in the data!



# Is training error a good measure of predictive performance?

Issue:

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



# Assessing the loss

## Part 2: Generalization (true) error

# Generalization error

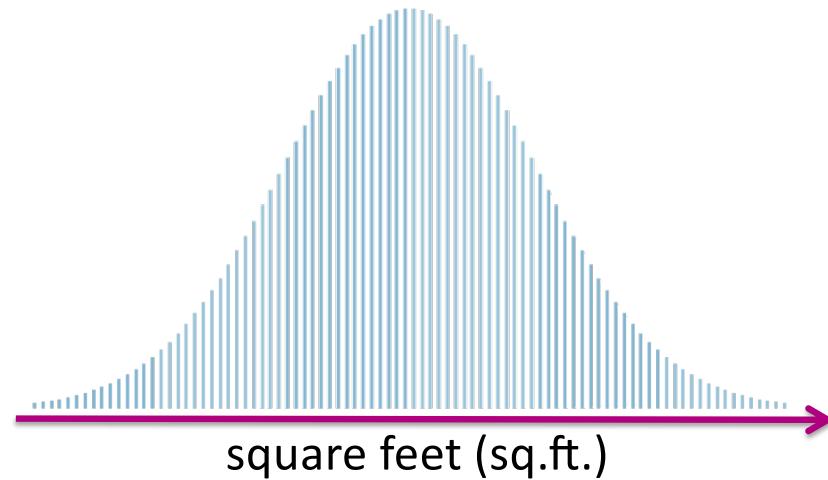
Really want estimate of loss over all possible (, ) pairs



# Distribution over houses

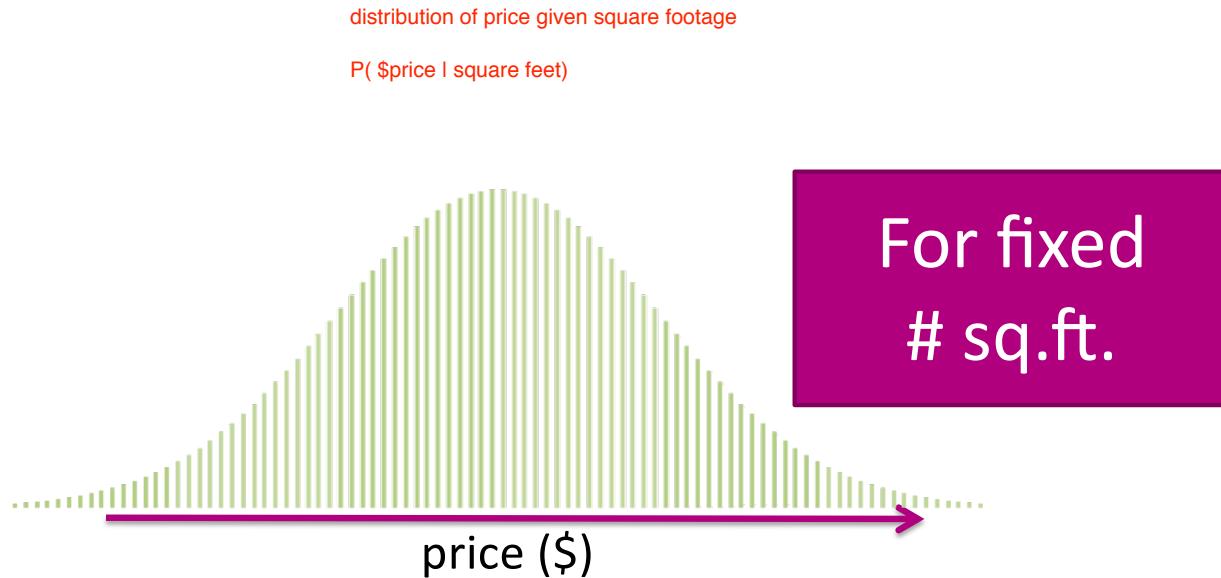
In our neighborhood, houses of what # sq.ft. () are we likely to see?

What is the variability in the square footage?



# Distribution over sales prices

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



# Generalization error definition

Using  $P(\text{house} = \text{sqrt feet})$  and  $P(\text{price} | \text{square feet})$  completely describes the distribution of all the pairs (square feet, price) we are interested in!

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

Formally:

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

( $x, y$ ) pairs our drawn from our true distribution  
 $p(x, y) = p(y | x)p(x)$ ... But how do we know this distribution?

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

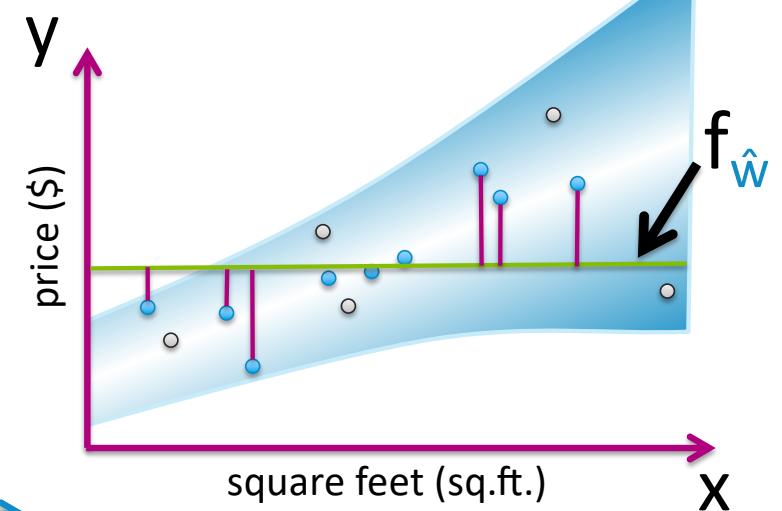
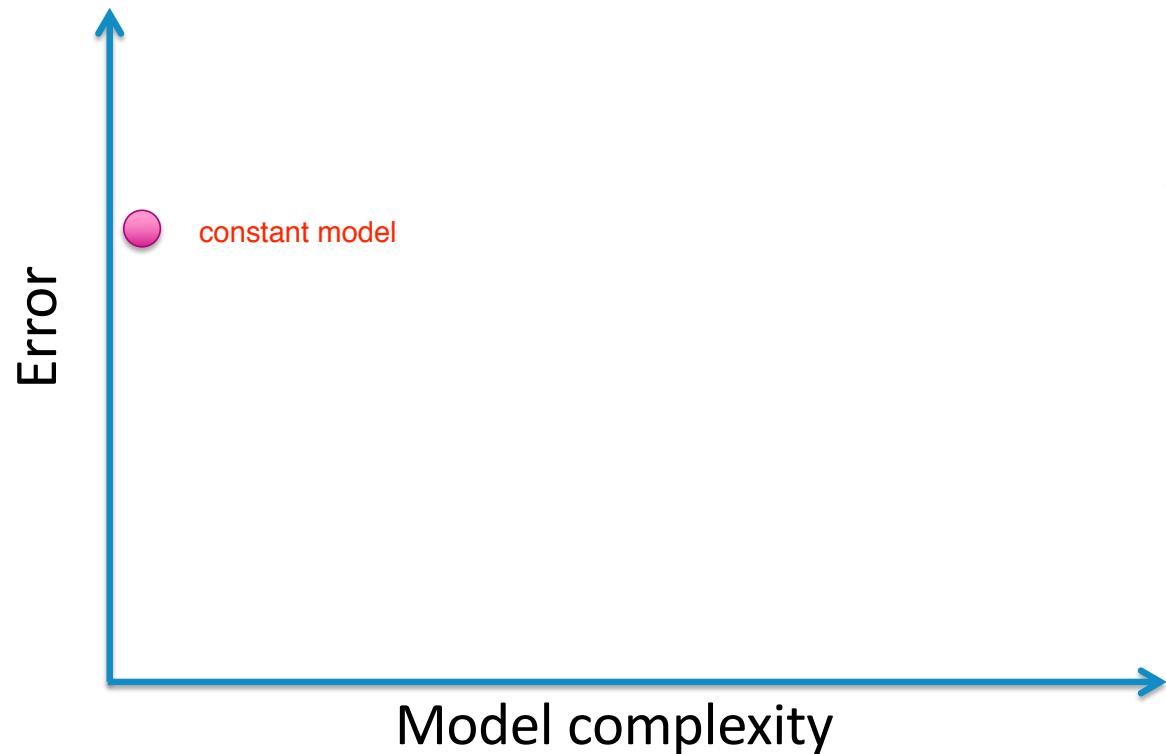
expected loss we have using this particular predictor function

$$\int_{(x,y)} (y - f(x))^2 p(x,y) dx$$

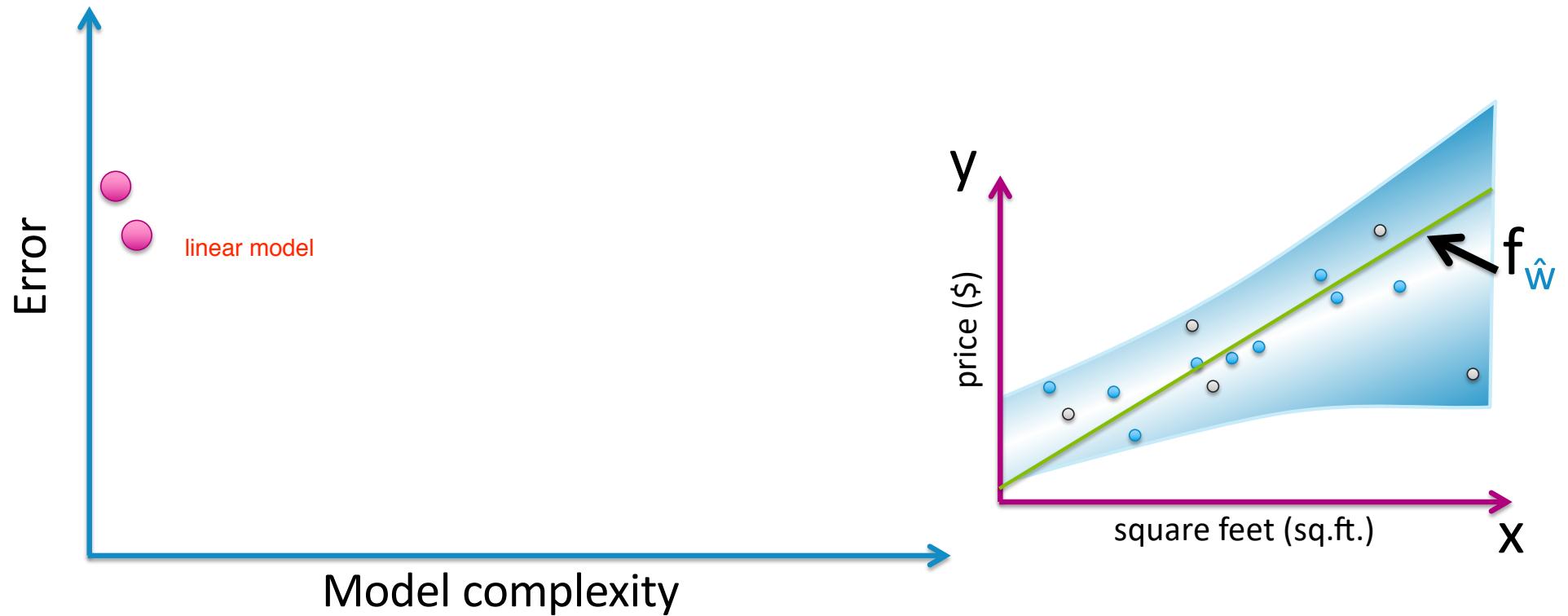


fit using training data

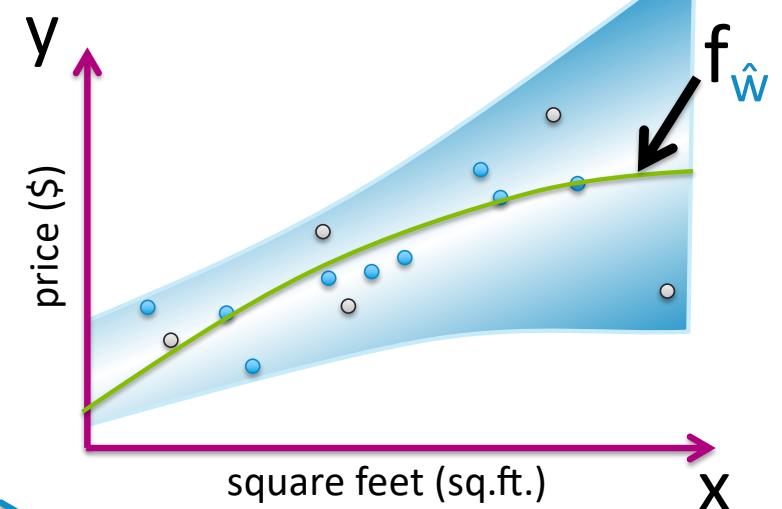
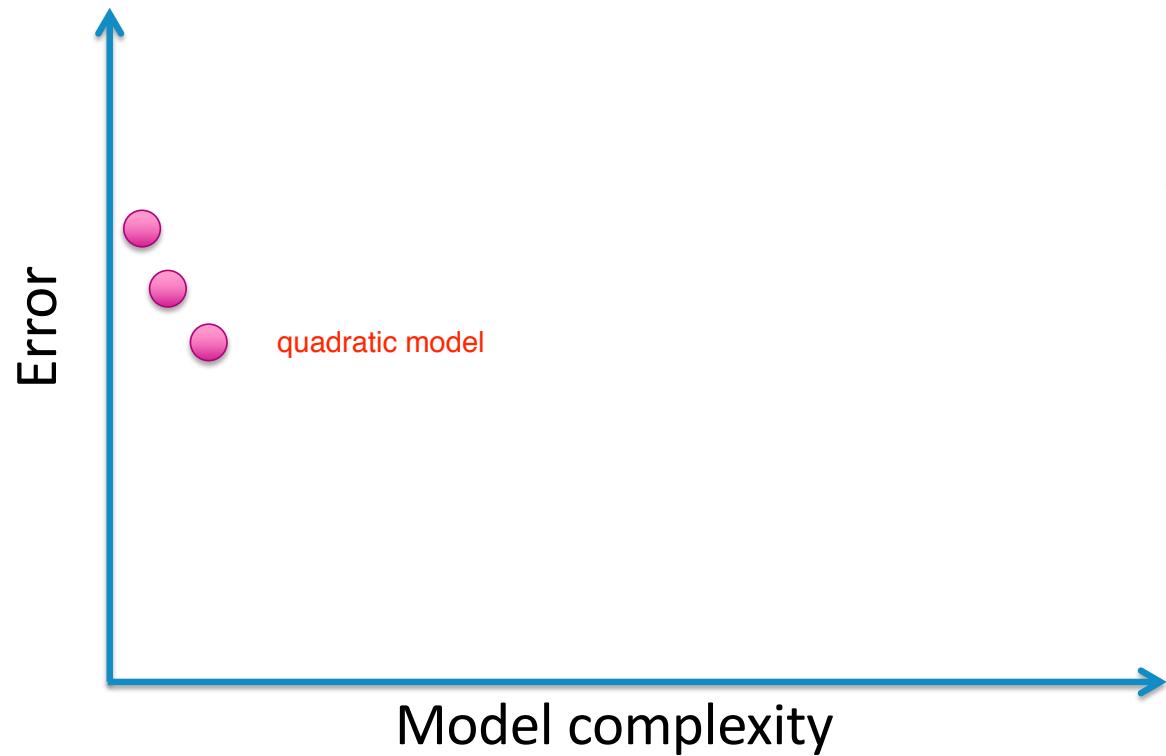
# Generalization error vs. model complexity



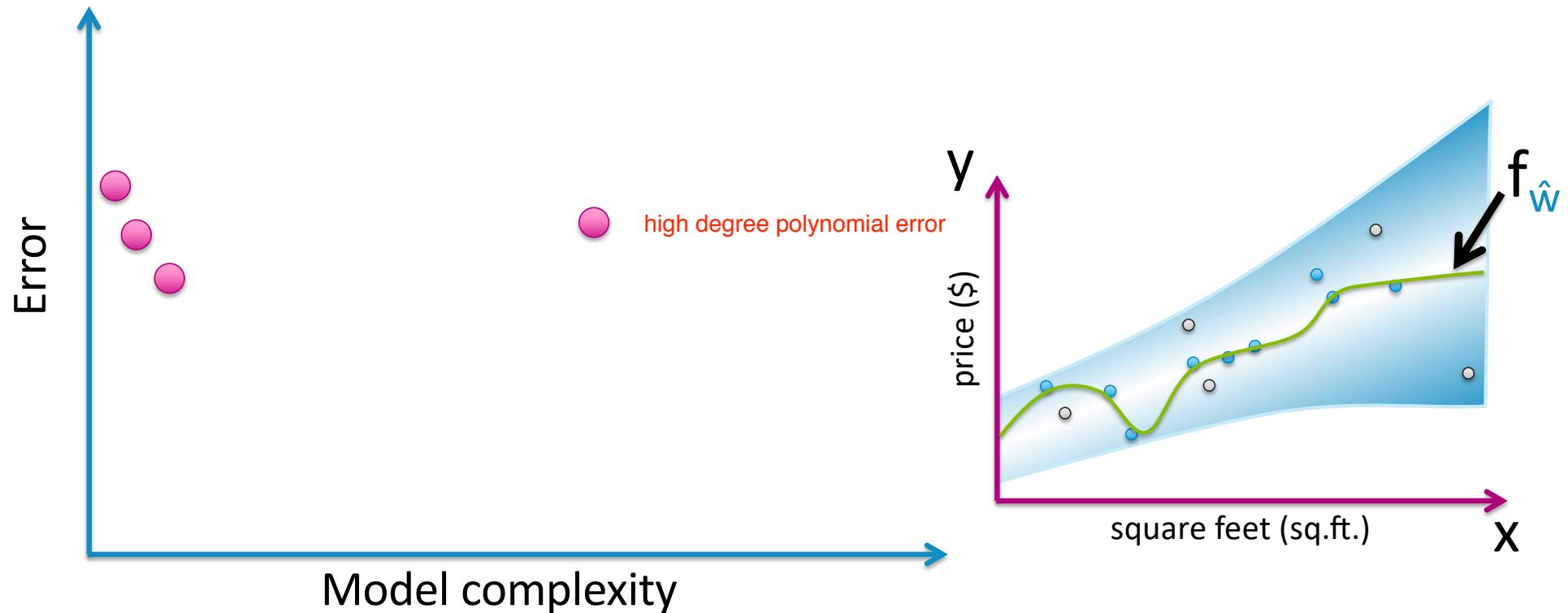
# Generalization error vs. model complexity



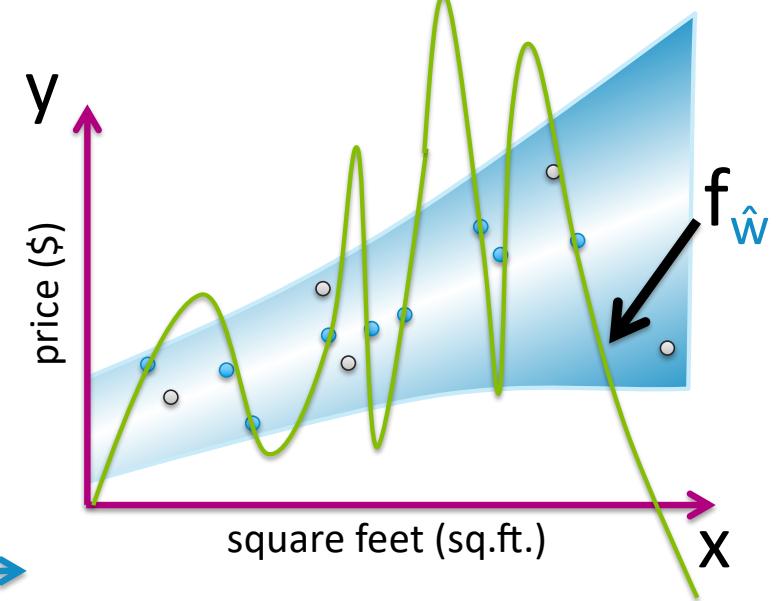
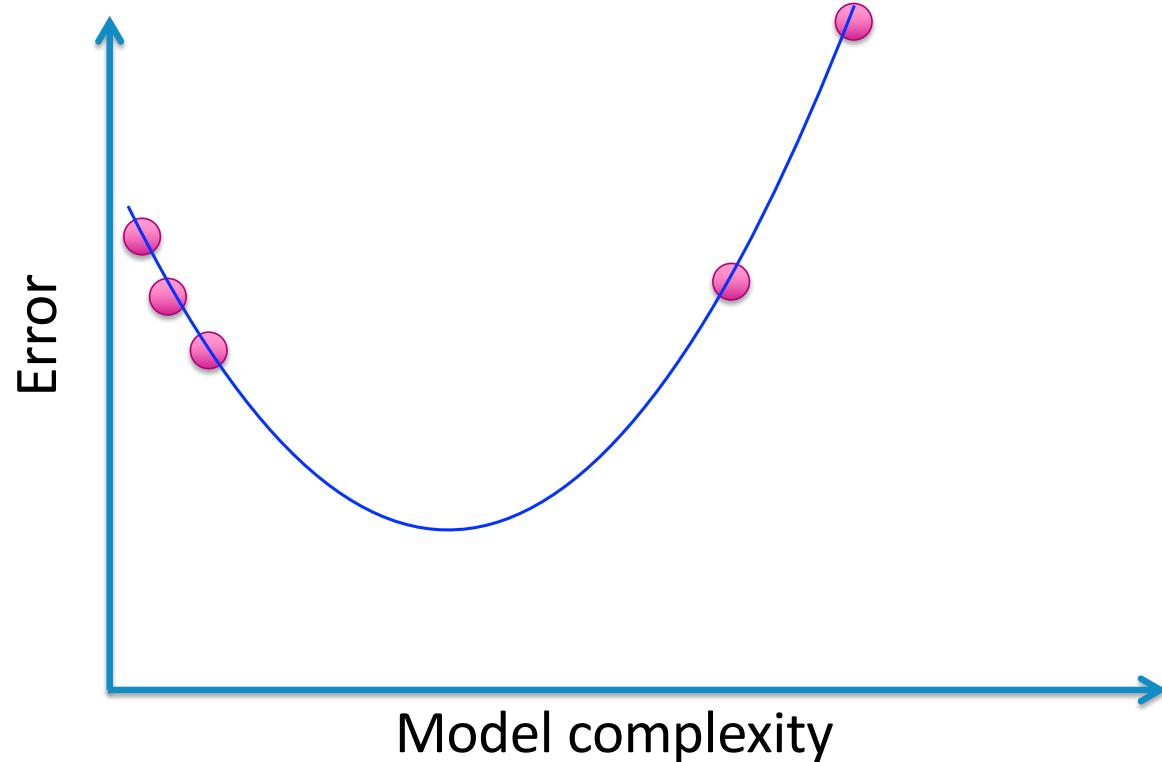
# Generalization error vs. model complexity



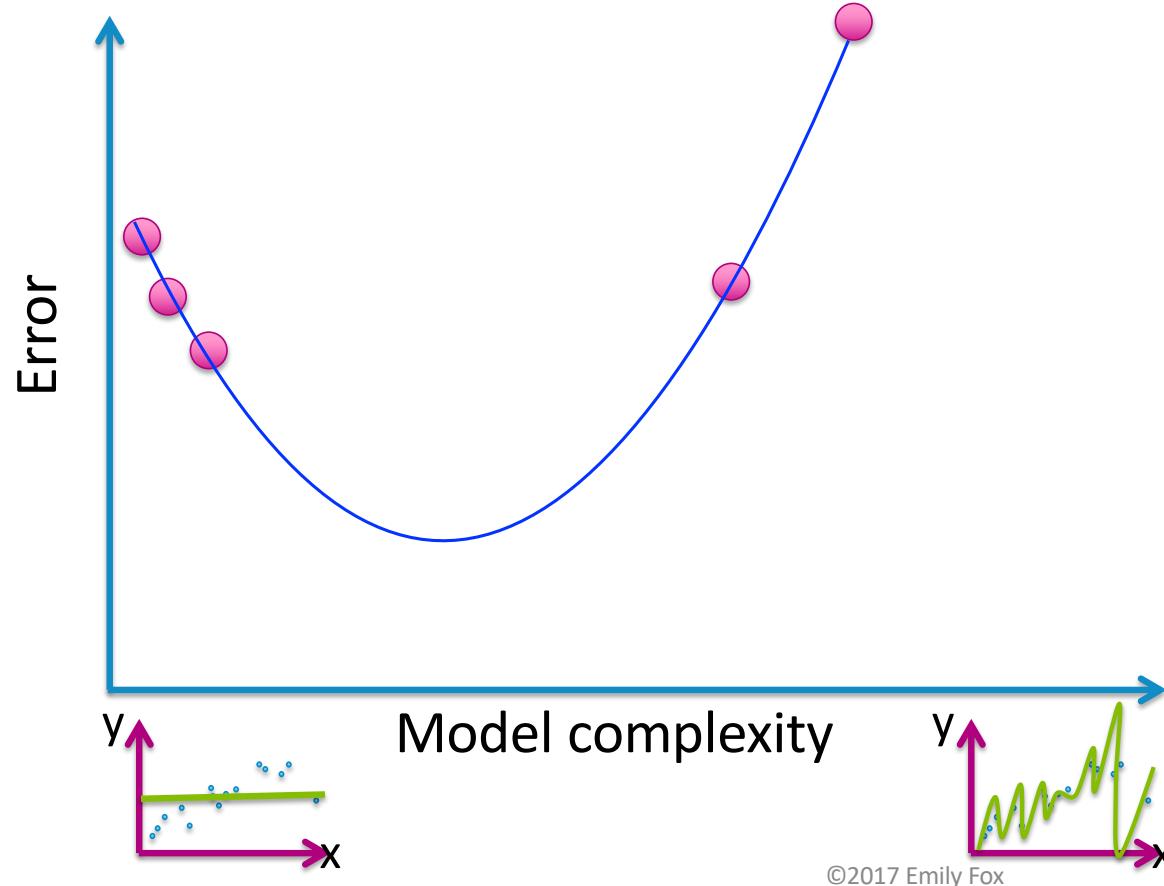
# Generalization error vs. model complexity



# Generalization error vs. model complexity



# Generalization error vs. model complexity



Can't  
compute!

do not know the distribution of  
 $p(x,y) = p(\text{price} \mid \text{square footage})p(\text{square footage})$

# 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**

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

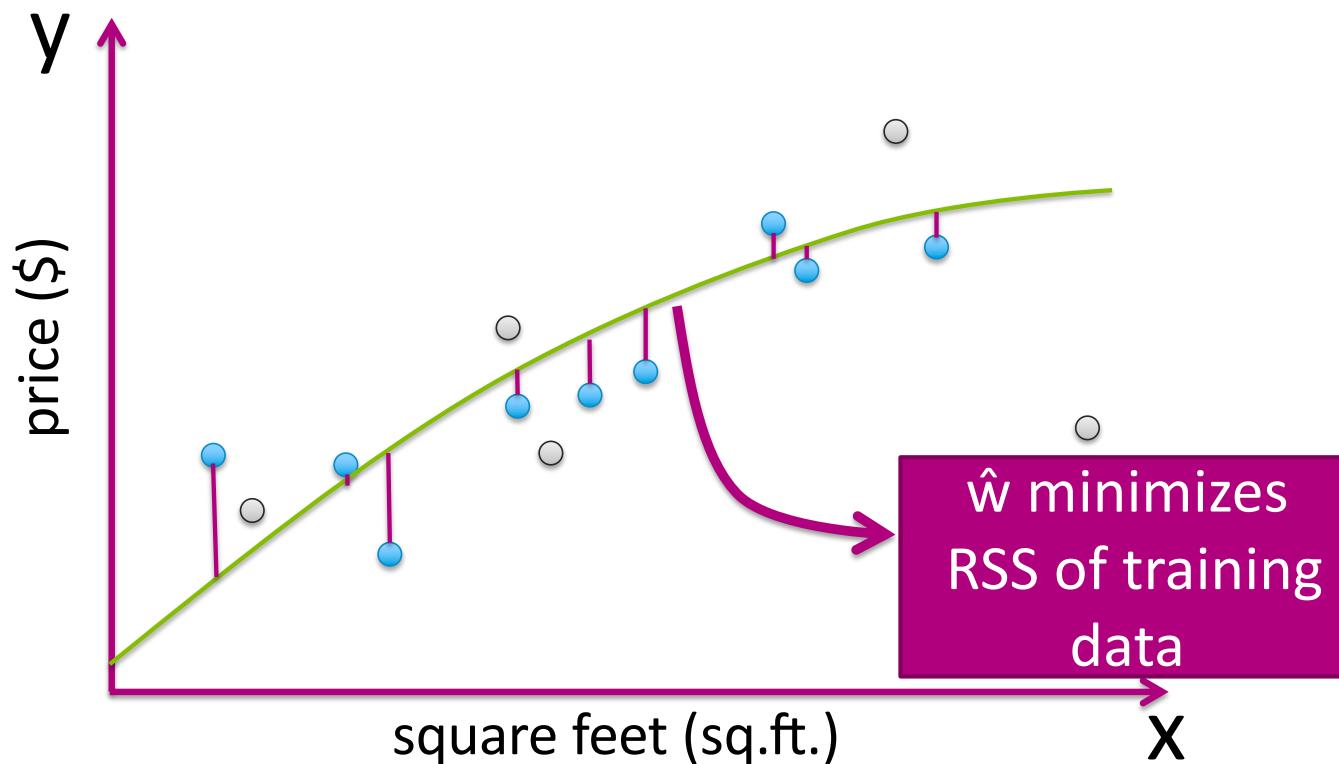
↑  
# test points

fit using **training data**

has never seen  
test data!

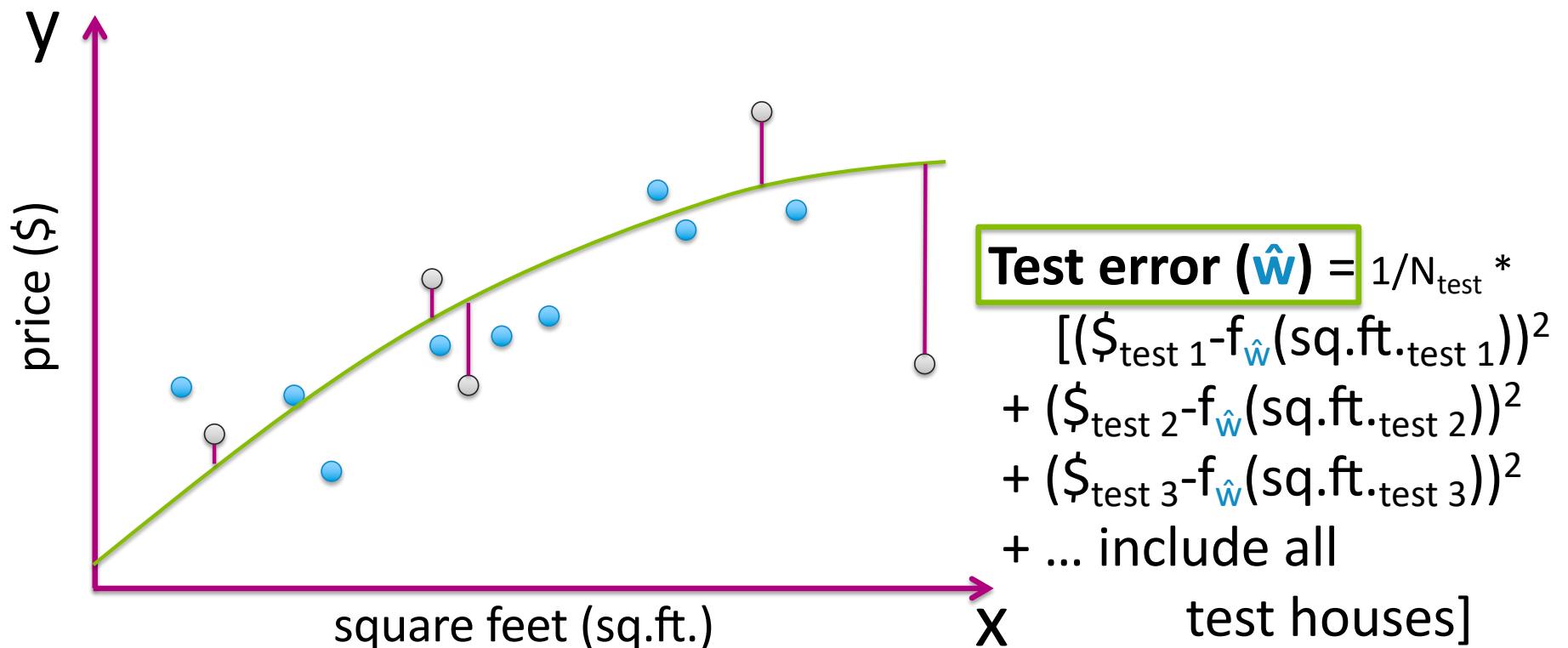
Example:

As before, fit quadratic to training data

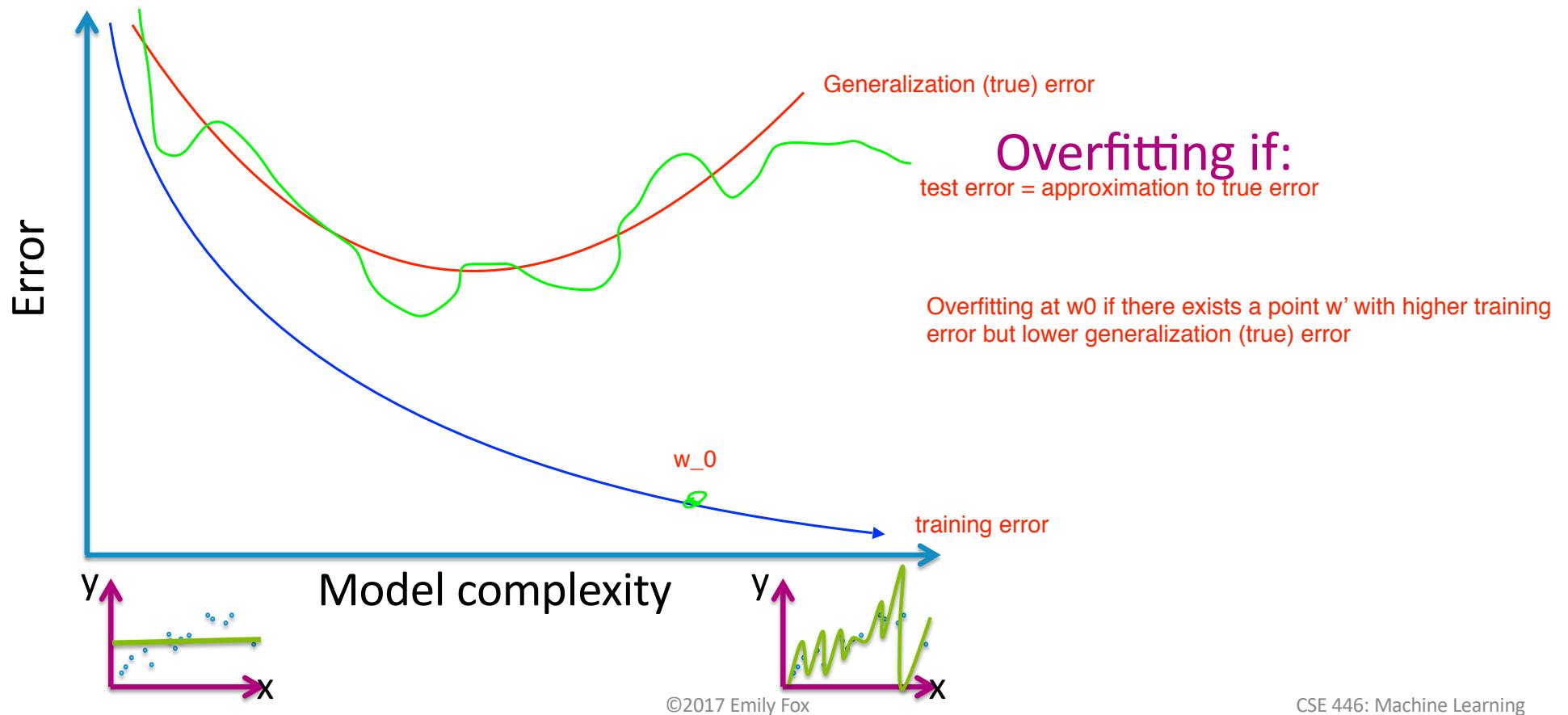


Assess performance using **test error**

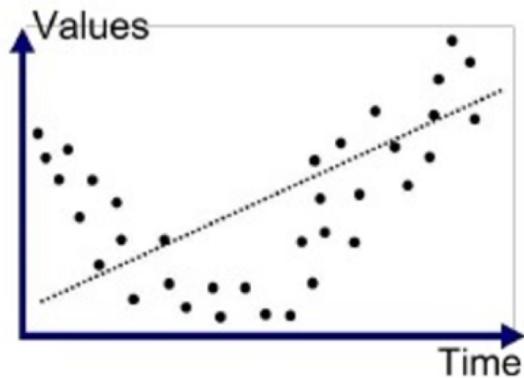
As before, use **squared error** loss  $(y - f_{\hat{w}}(x))^2$



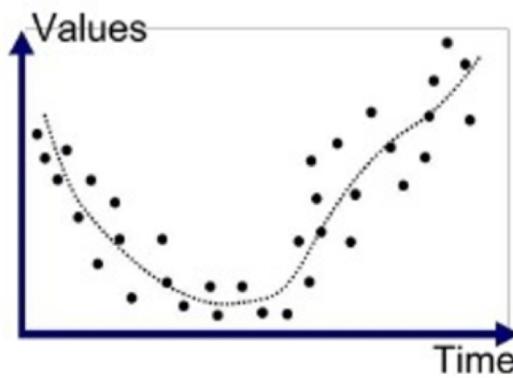
# Training, true, & test error vs. model complexity



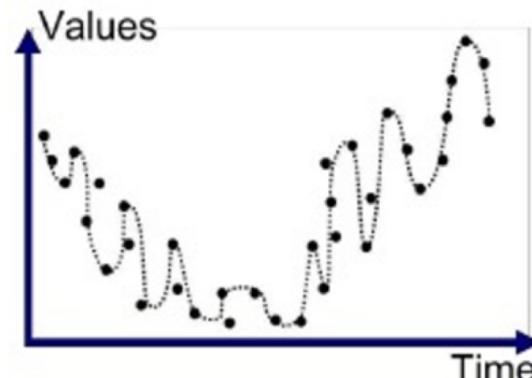
# Underfitting vs Overfitting



Underfitted



Good Fit/R robust

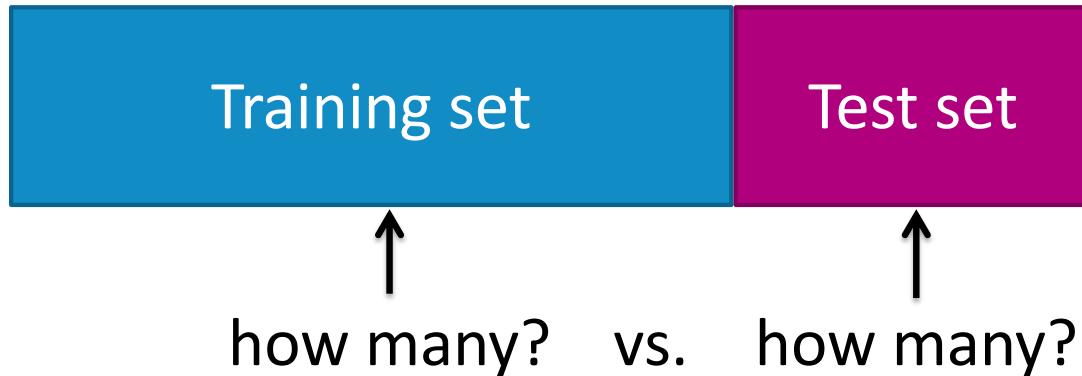


Overfitted

Courtesy Blog@AlgoTrading101

# *Training/test split*

# Training/test splits

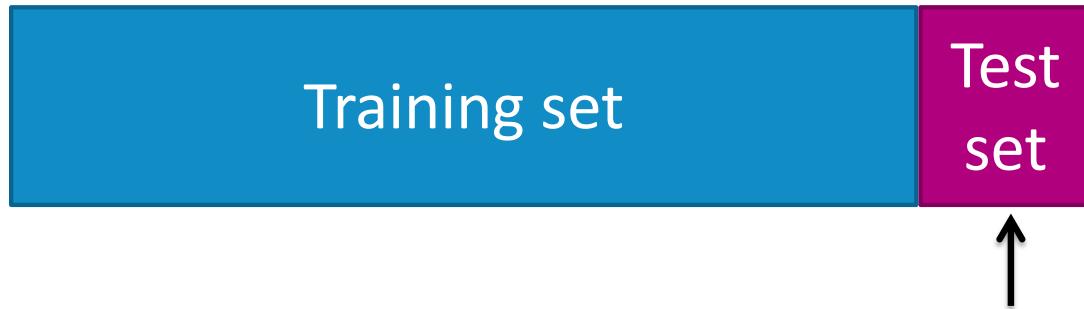


# Training/test splits



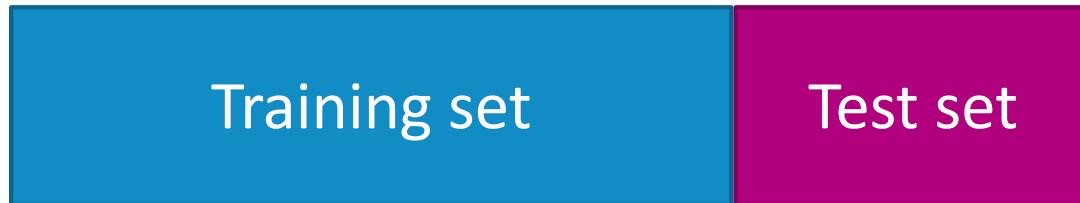
Too few  $\rightarrow \hat{w}$  poorly estimated

# 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

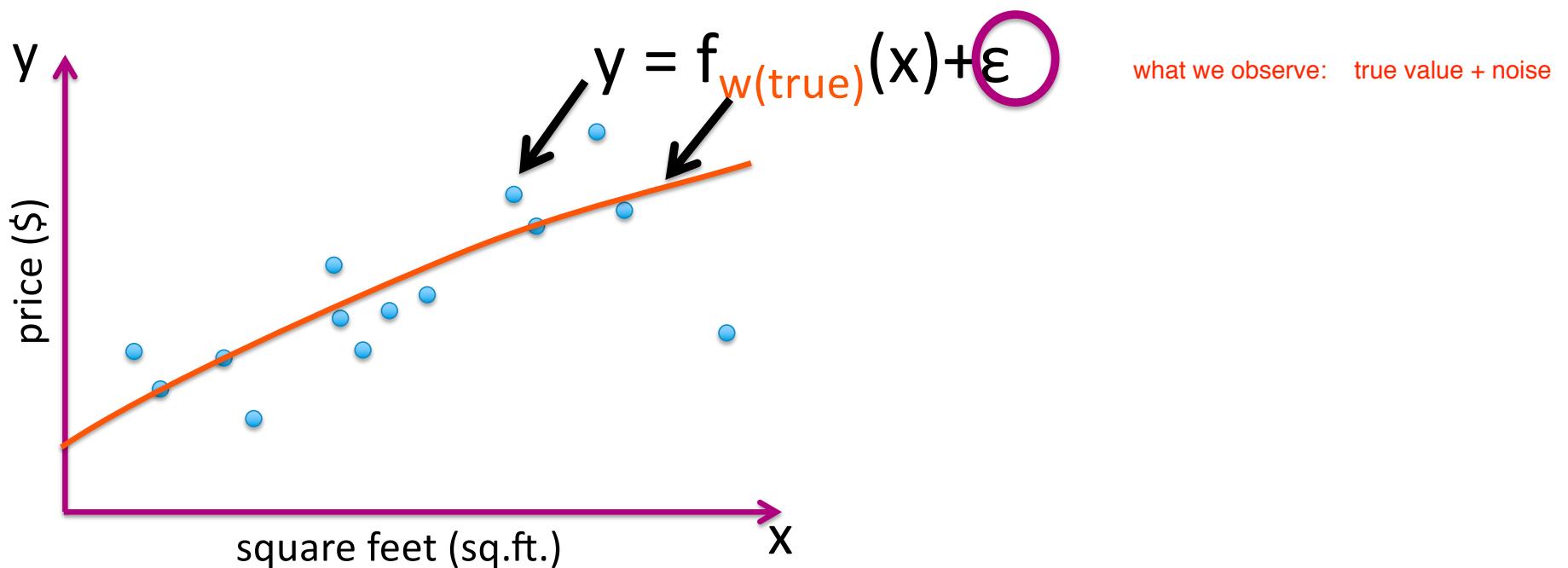
# 3 sources of error

In forming predictions, there are 3 sources of error:

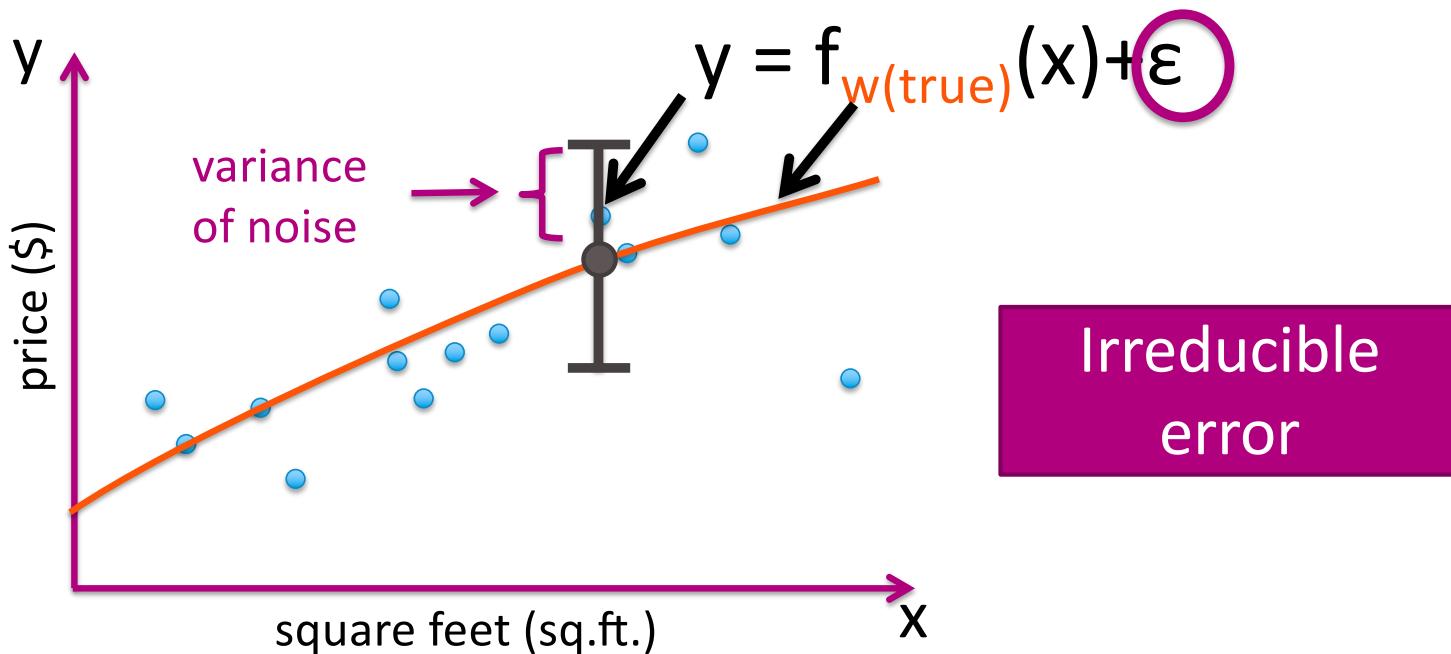
1. Noise
2. Bias
3. Variance

# Data inherently noisy

natural variations in data (due to unseen or random forces)

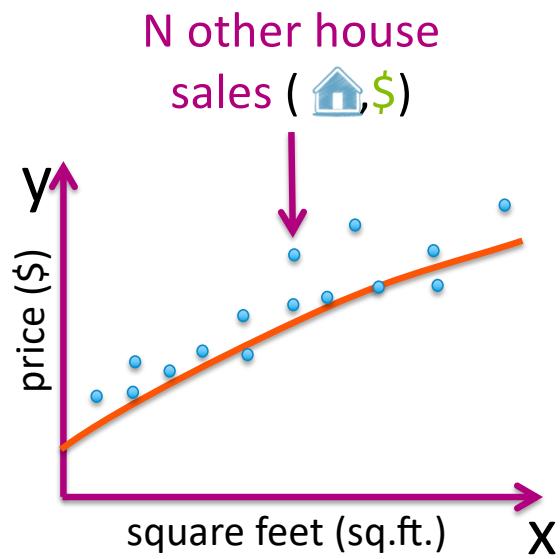
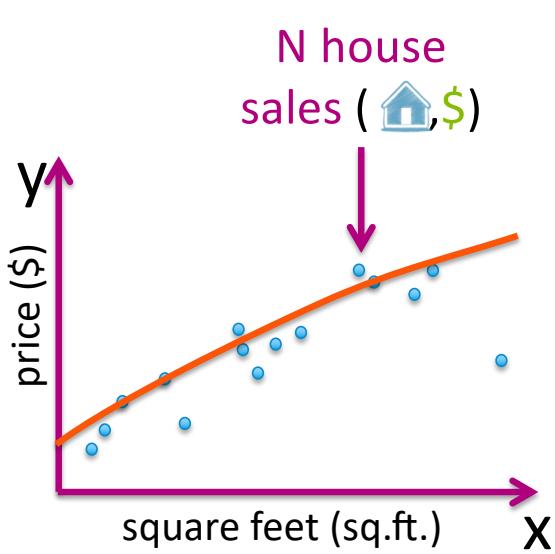


# Data inherently noisy



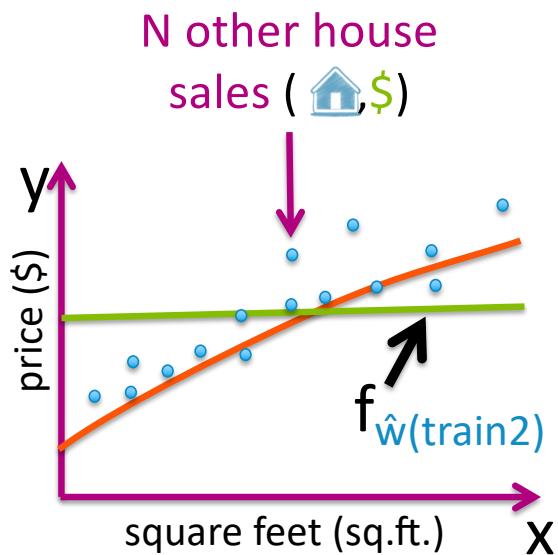
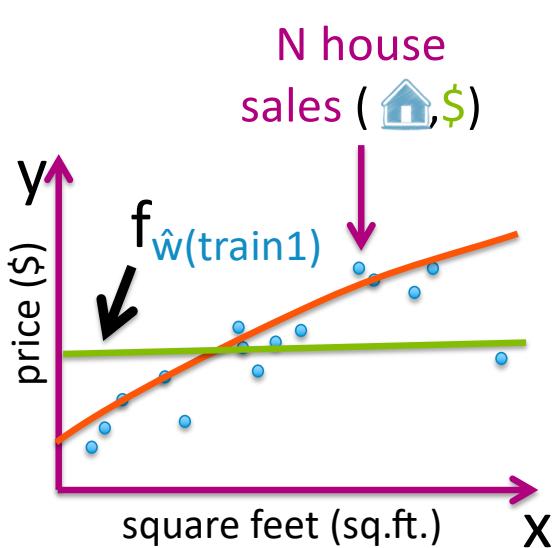
# Bias contribution

Assume we fit a constant function



# Bias contribution

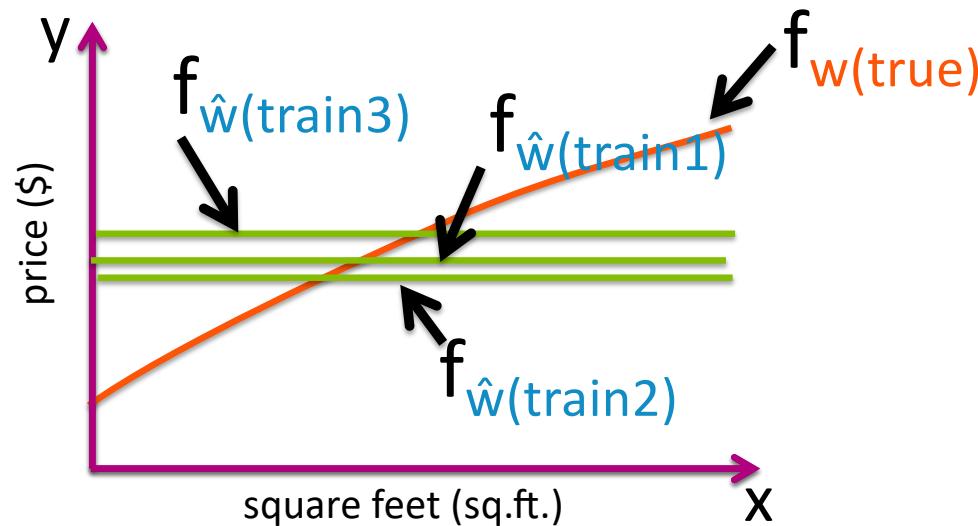
Assume we fit a constant function



## Bias contribution

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

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

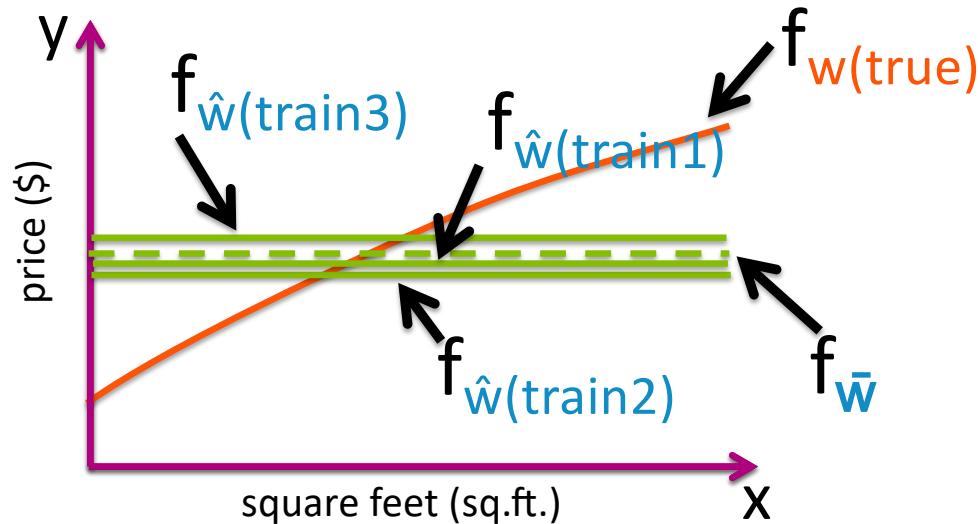


If we knew the true predictor for the data, how well would our model be able to capture it?

## Bias contribution

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

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

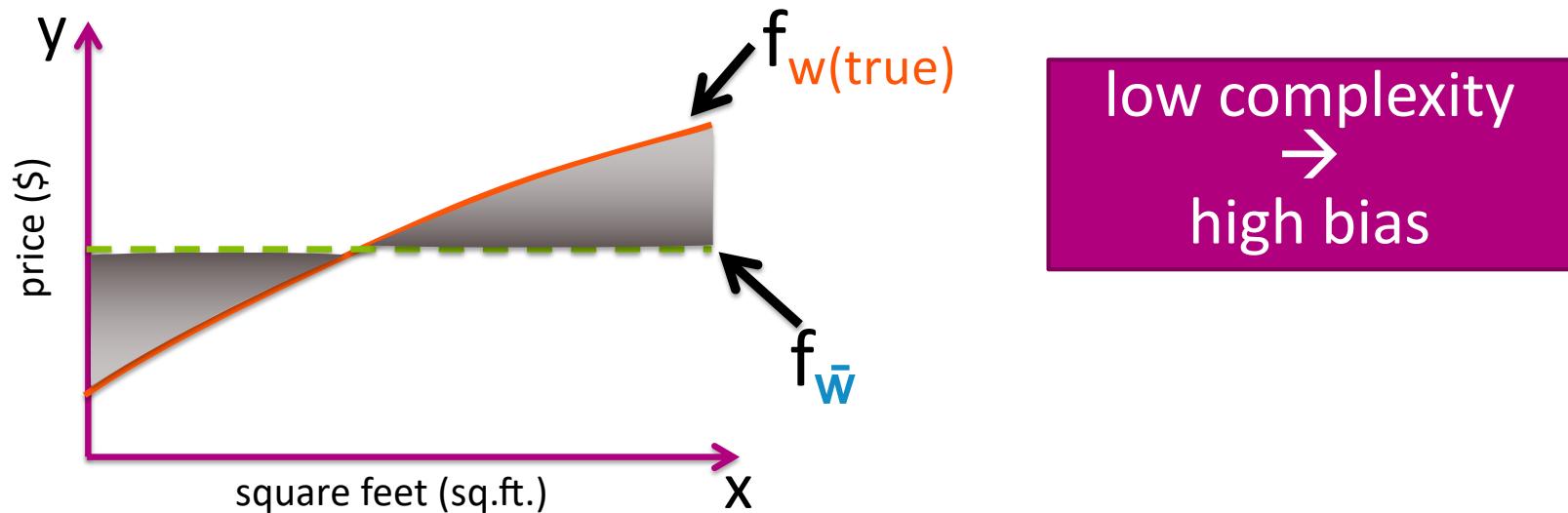


# Bias contribution

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

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

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



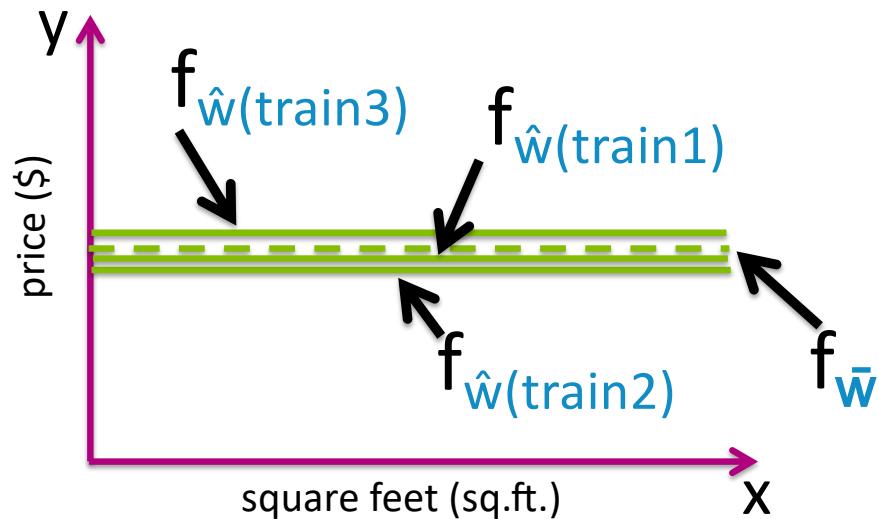
$$f_{\bar{w}}(x) = E_{\text{train}} f[\hat{w}(\text{train})(x)]$$

## Variance contribution

How much do specific fits vary from the expected fit?

variance in the model that we get. How much would noise effect the parameters of our model?  
If we resampled our data would we get very similar parameters or would we get something vastly different?

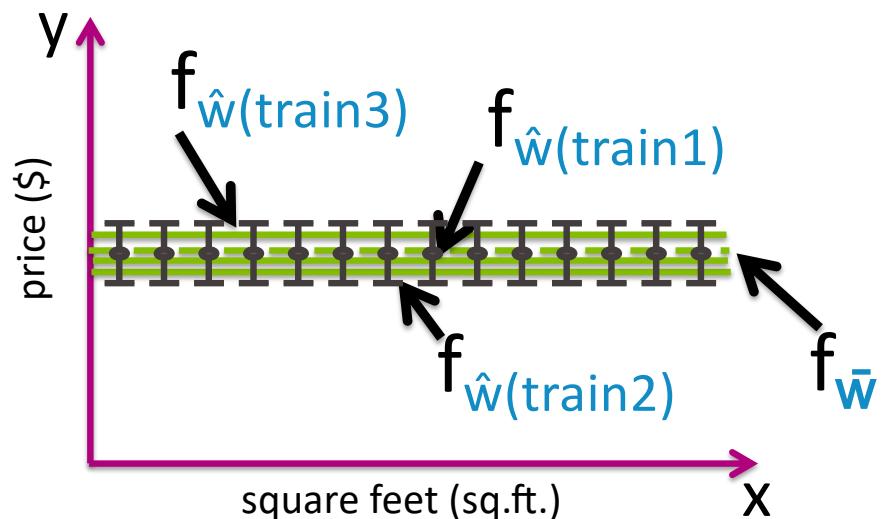
Typically less variance for less complex models.



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

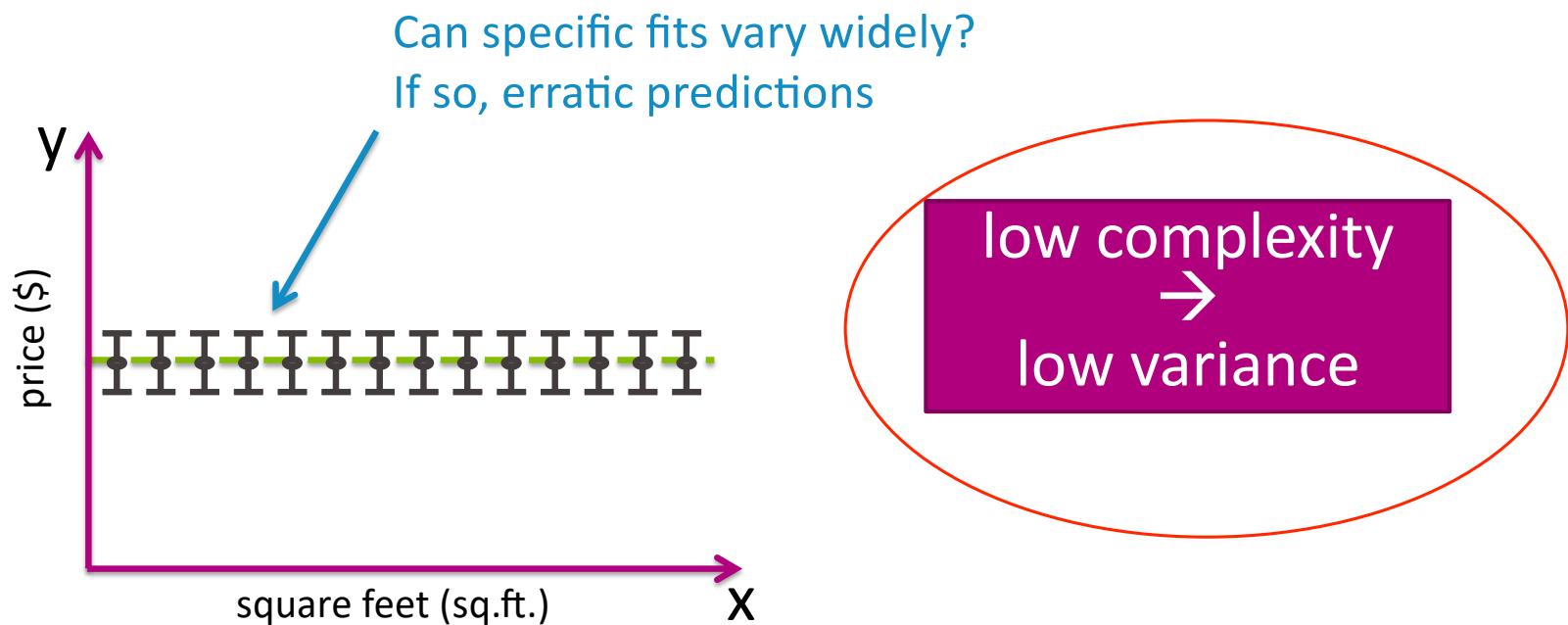
## Variance contribution

How much do specific fits vary from the expected fit?



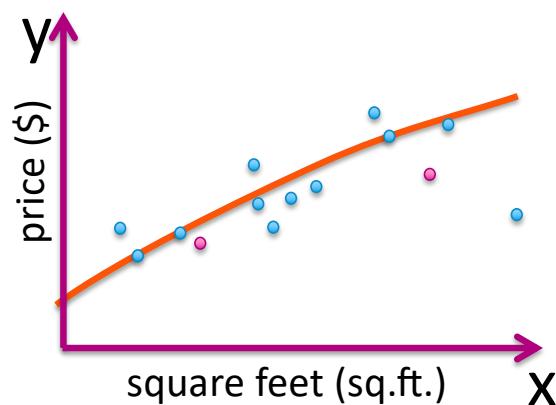
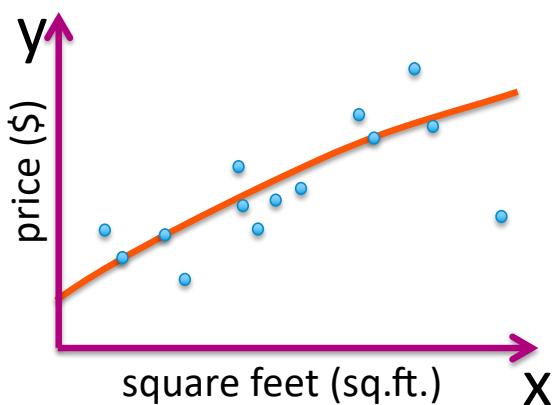
# Variance contribution

How much do specific fits vary from the expected fit?



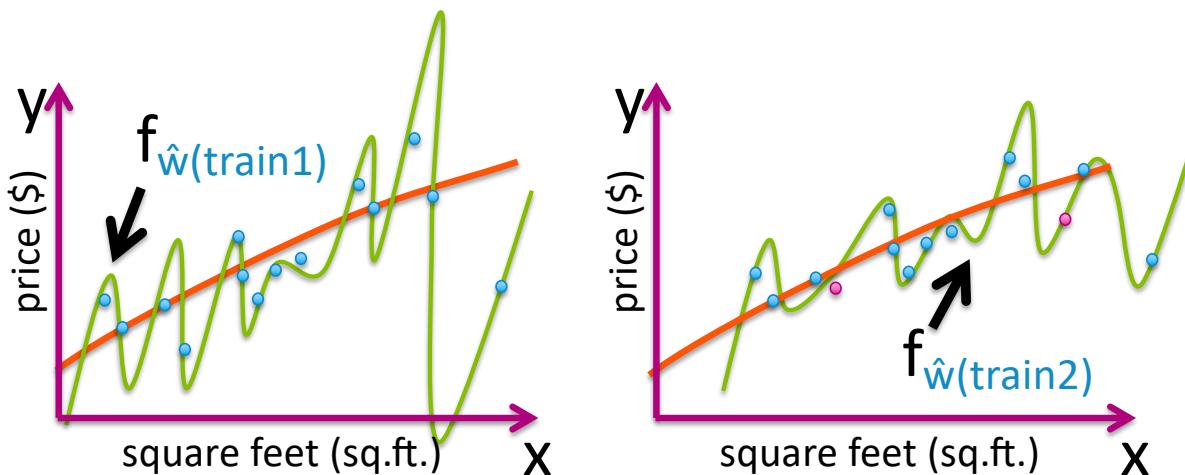
# Variance of high-complexity models

Assume we fit a high-order polynomial



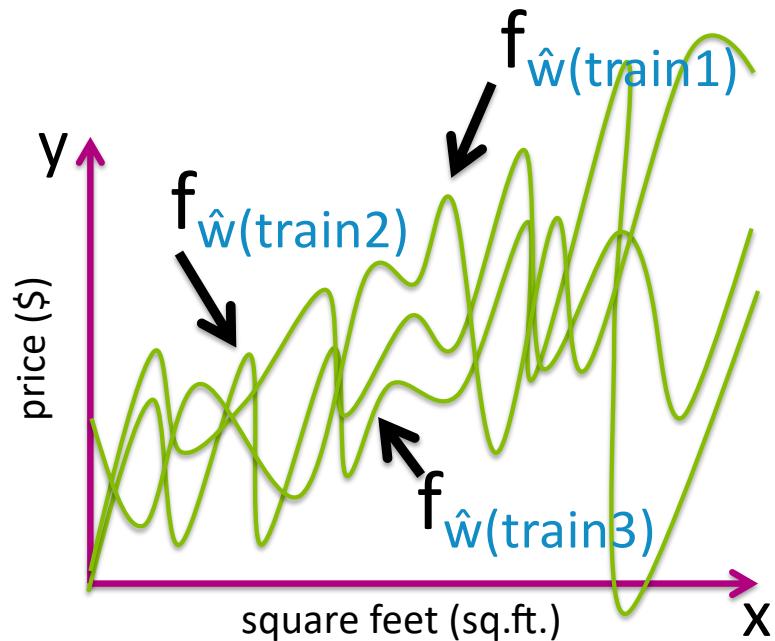
# Variance of high-complexity models

Assume we fit a high-order polynomial



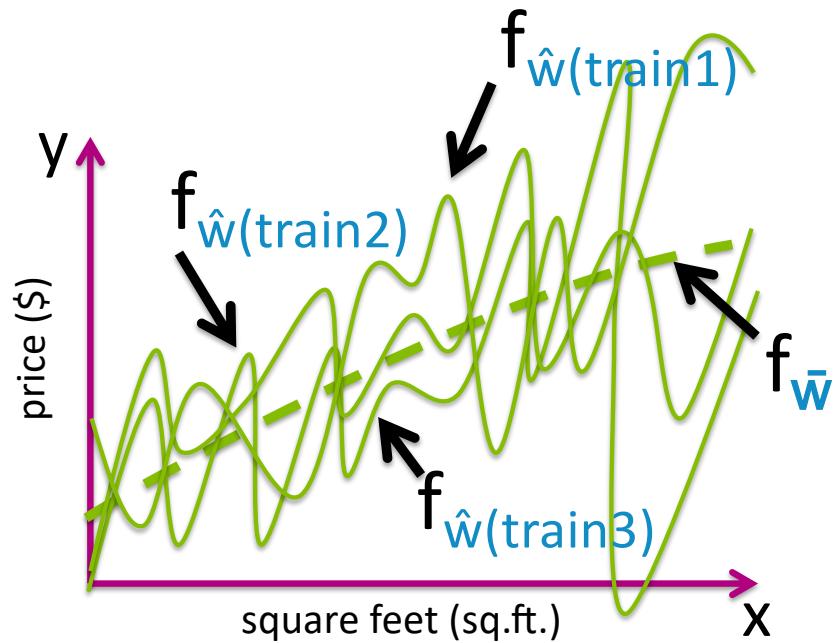
# Variance of high-complexity models

Assume we fit a high-order polynomial

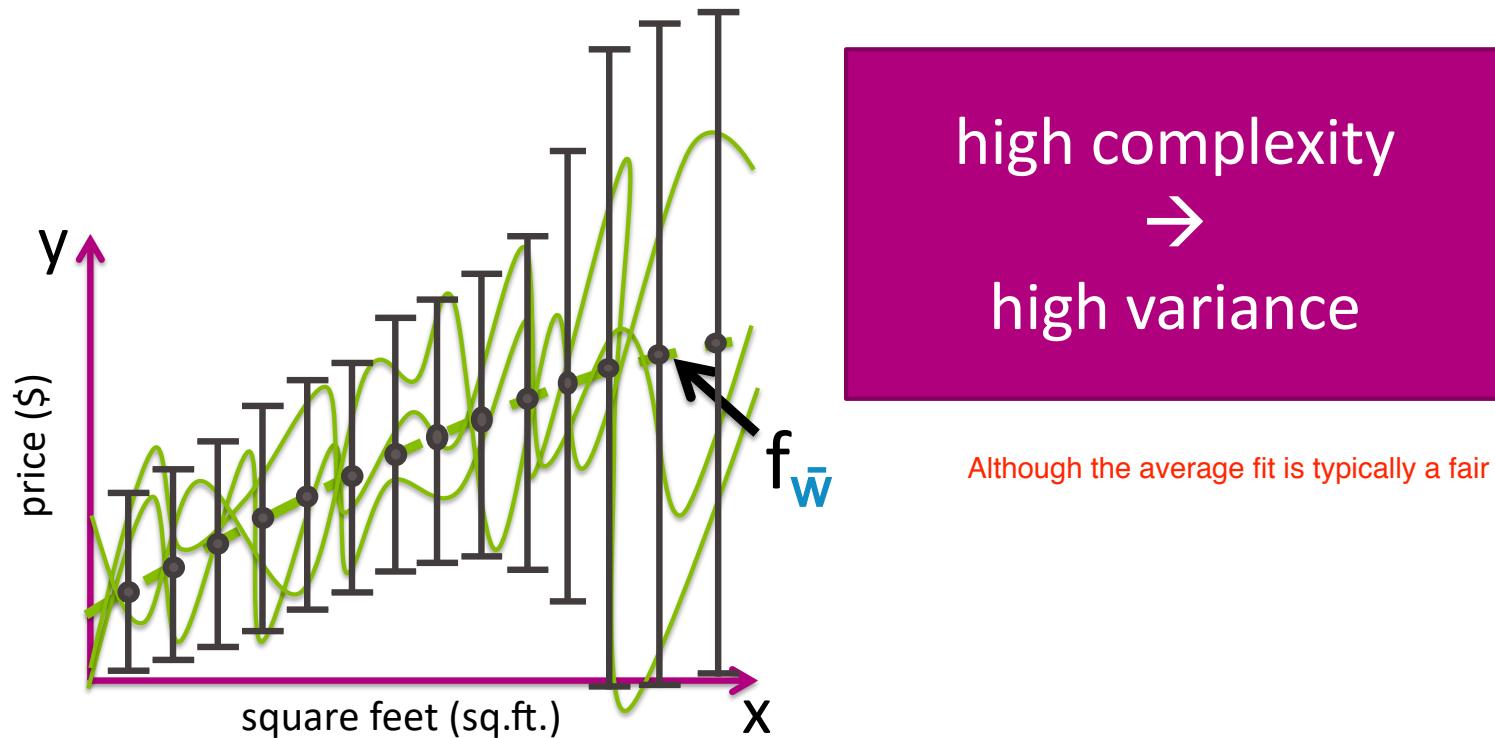


# Variance of high-complexity models

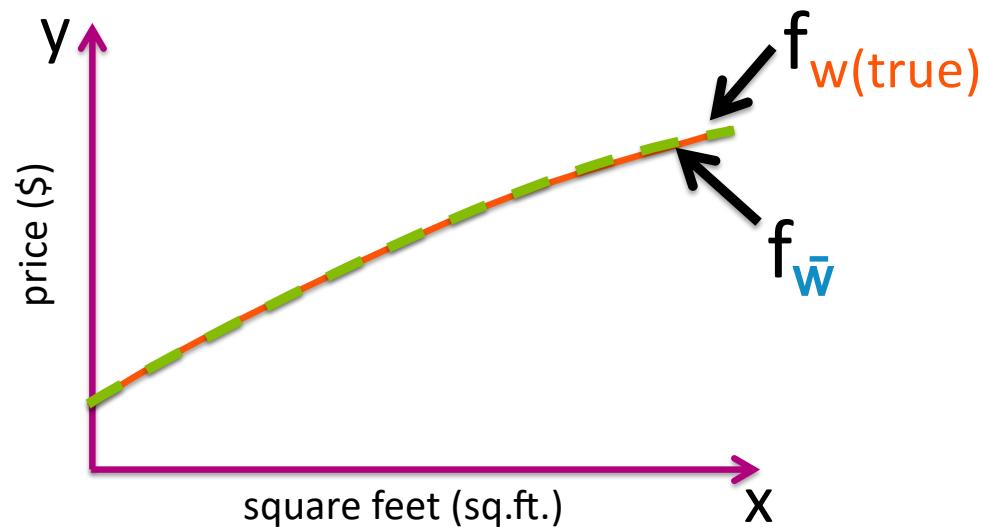
Assume we fit a high-order polynomial



# Variance of high-complexity models



# Bias of high-complexity models

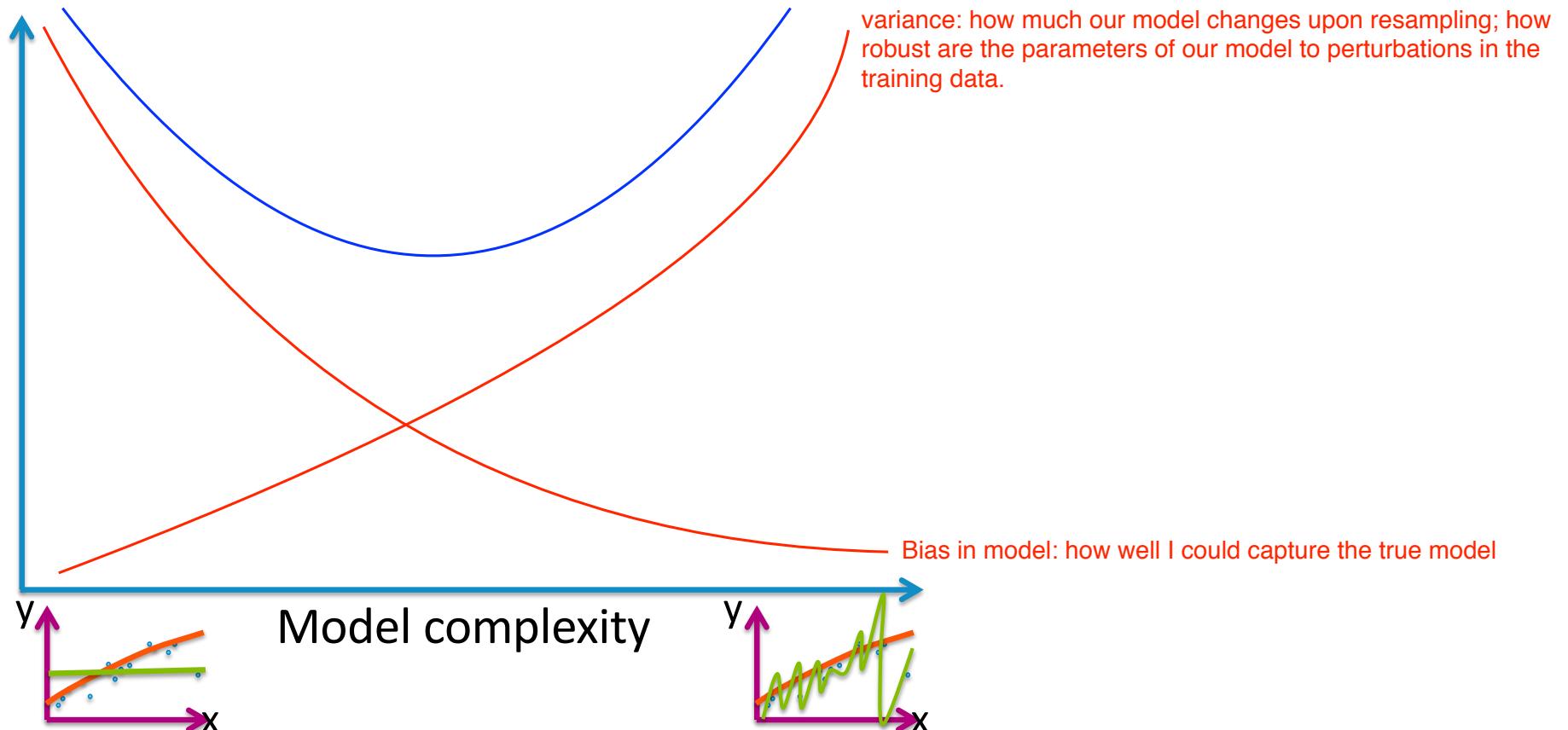


high complexity  
→  
low bias

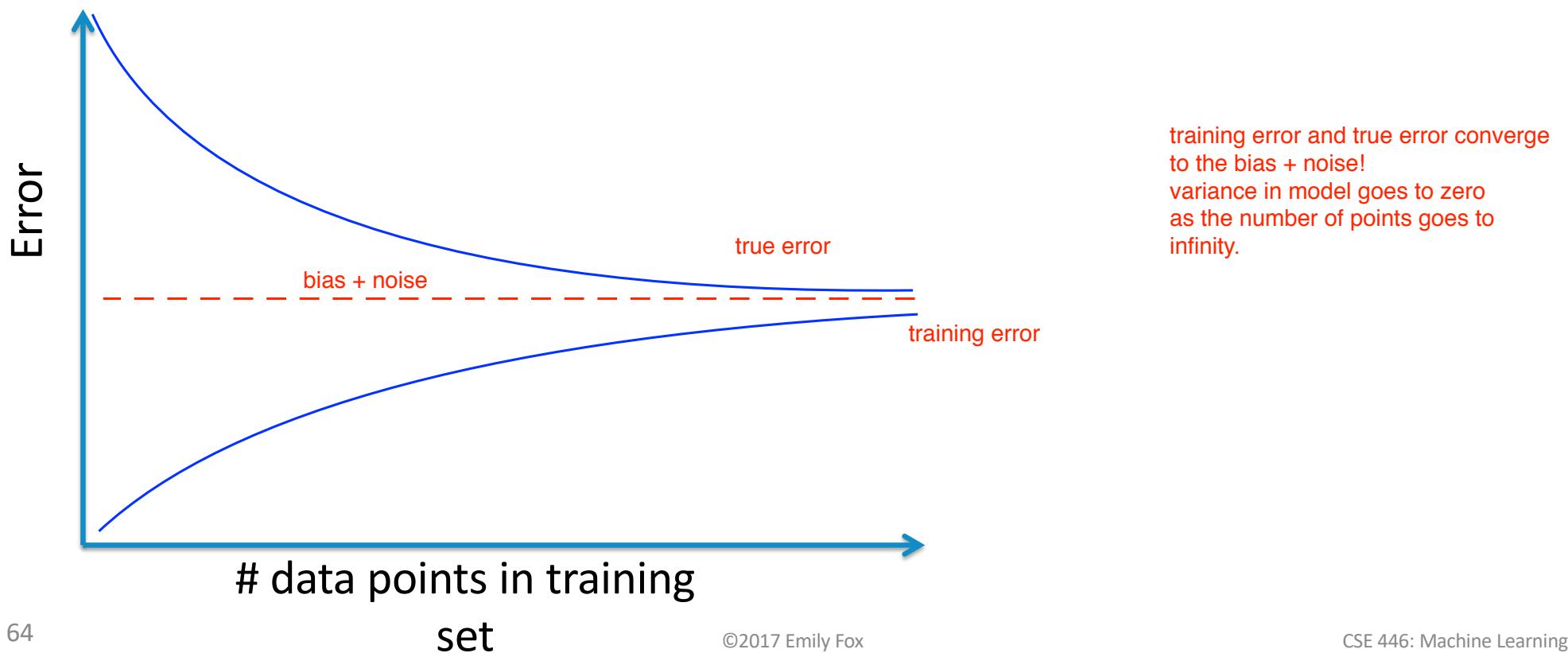
COULD capture the true model very well.

# Bias-variance tradeoff

$$MSE = \text{bias}^2 + \text{variance}$$



# True error and training error vs. amount of data for fixed model complexity

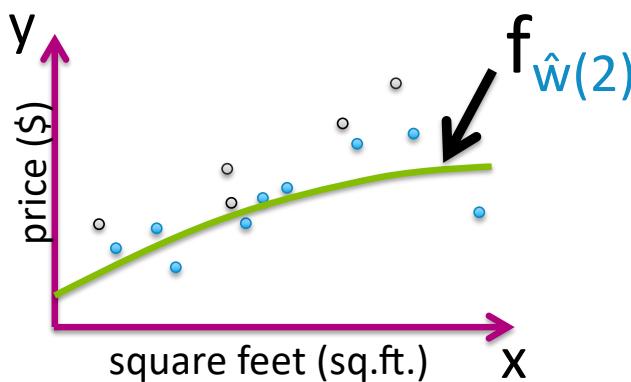
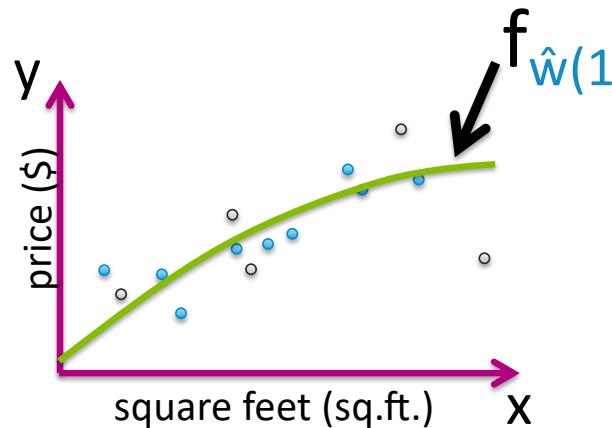


# Formalize 3 sources of errors...

# Accounting for training set randomness

Training set was just a random sample of  $n$  houses sold

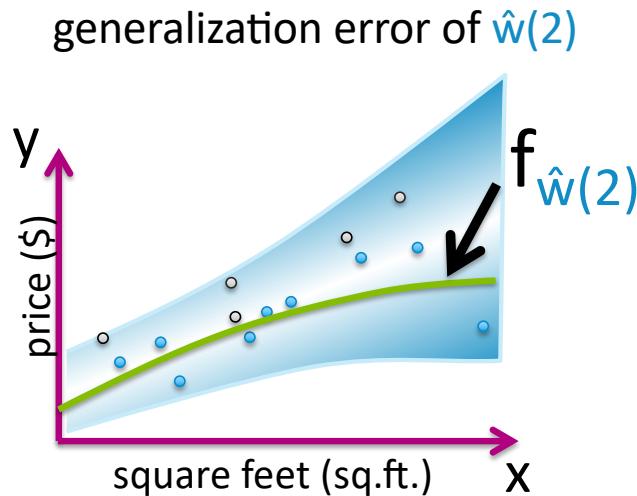
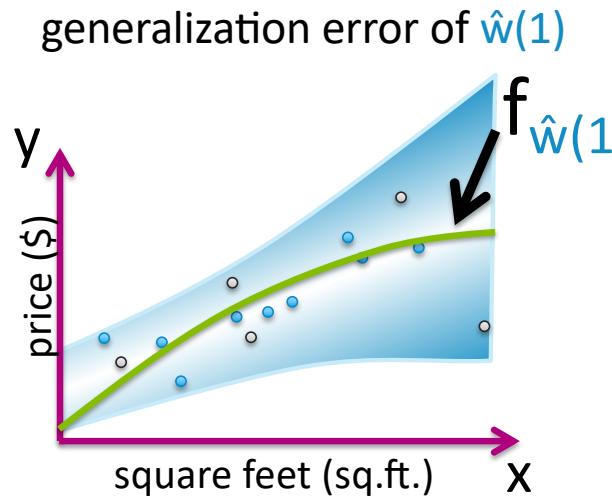
What if  $n$  other houses had been sold and recorded?



# Accounting for training set randomness

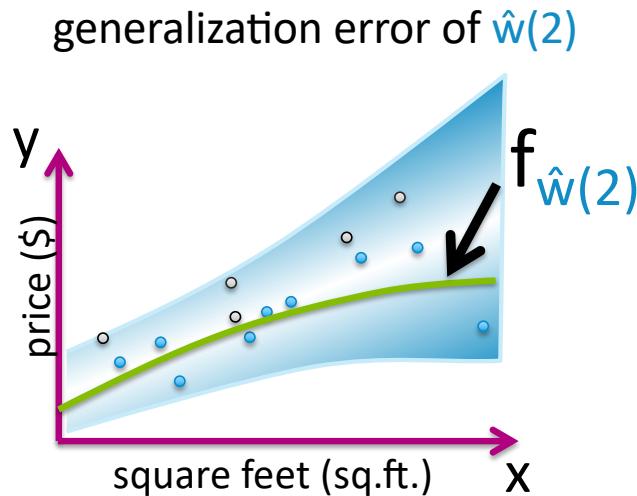
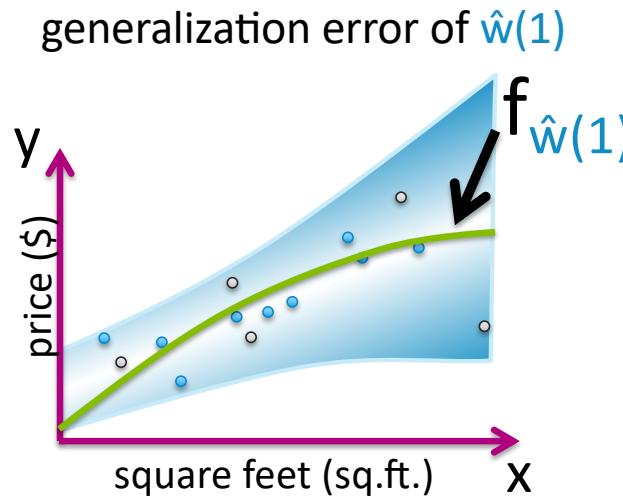
Training set was just a random sample of  $n$  houses sold

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

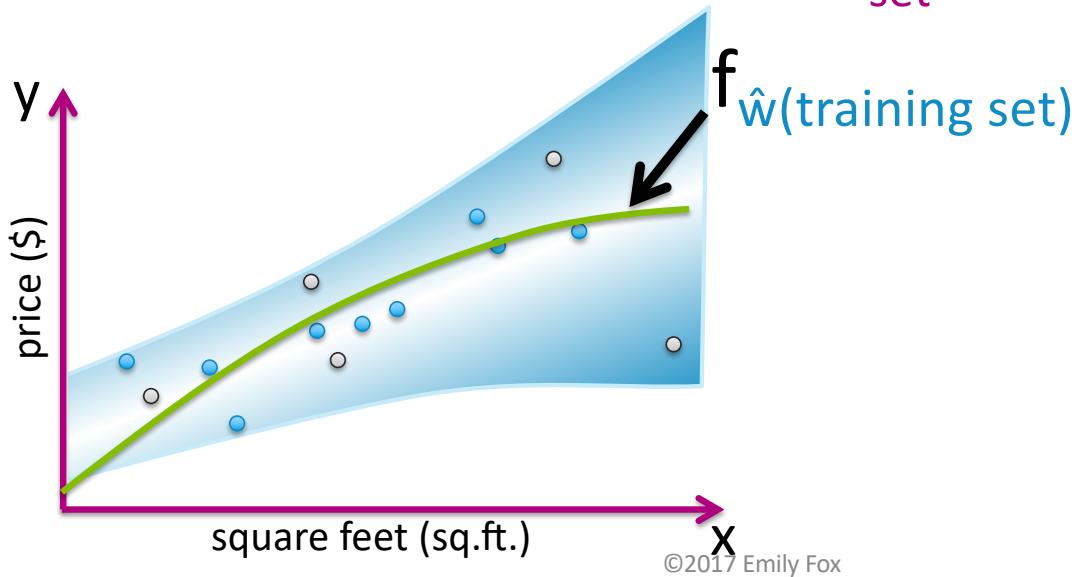


# Expected prediction error

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

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

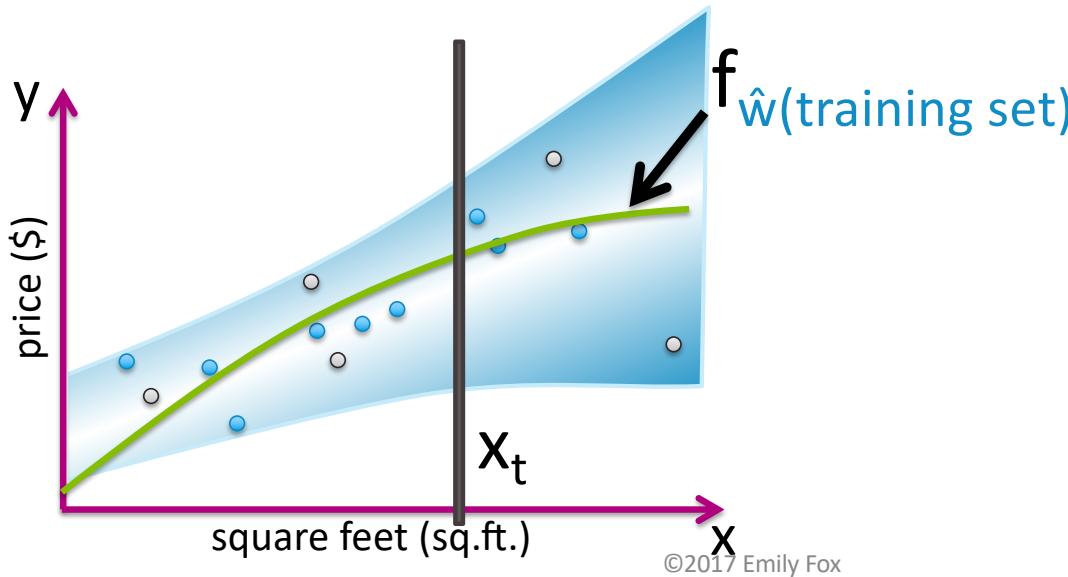
↑  
parameters fit on  
a specific training  
set



# Prediction error at target input

Start by considering:

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

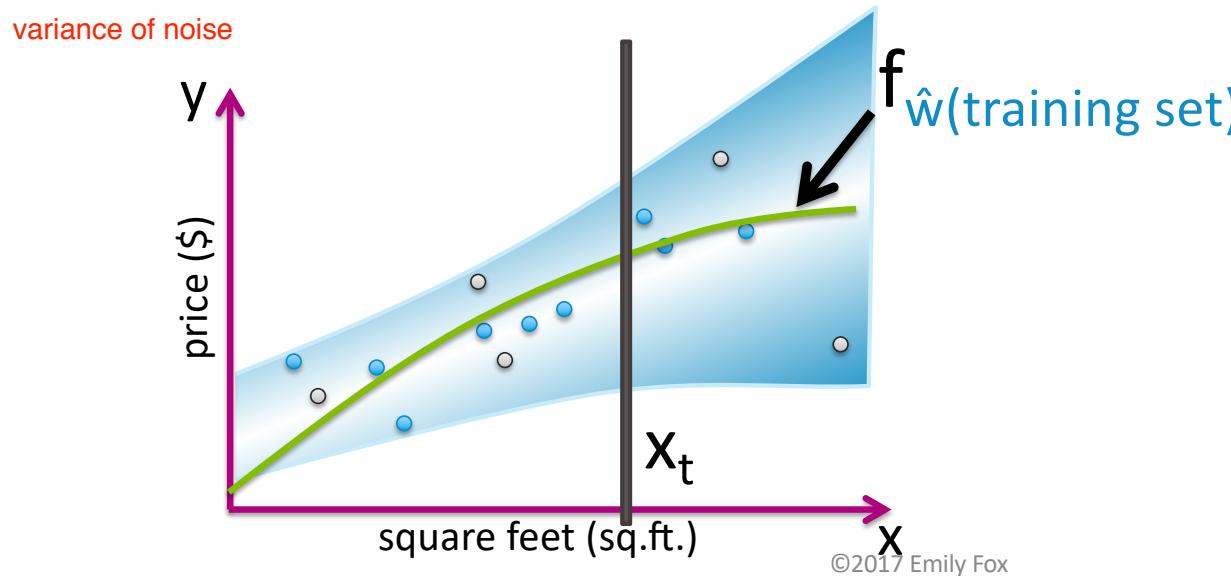


# Sum of 3 sources of error

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 noise  
Bias how well on avg  
model can fit  $f_{\hat{w}}$  ( $w$  true)  
Variance variability of fit

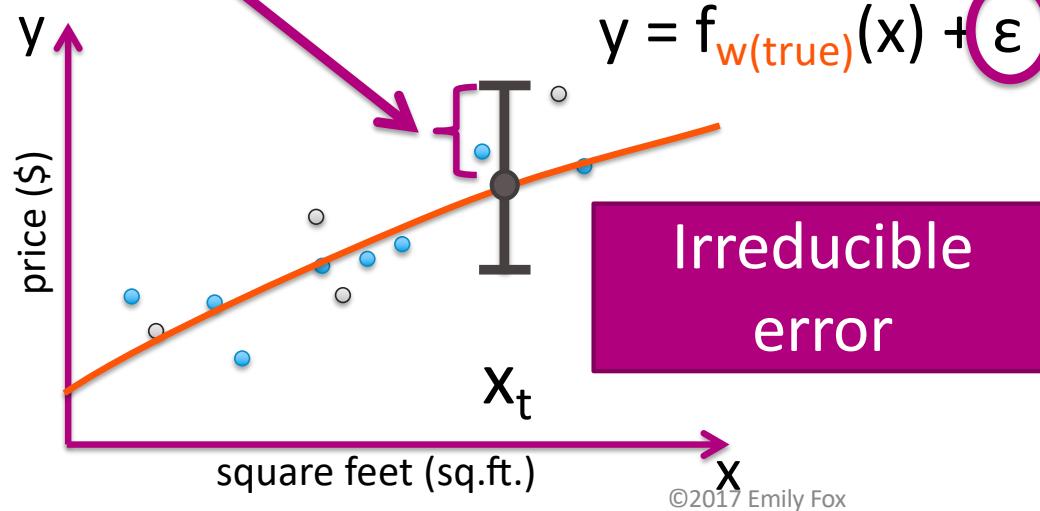


# Error variance of the model

$$\text{Variance of noise at } x_t = E_{y|x_t} [(y - f_{w(\text{true})}(x_t))^2] = \sigma^2$$

Average prediction error at  $x_t$

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

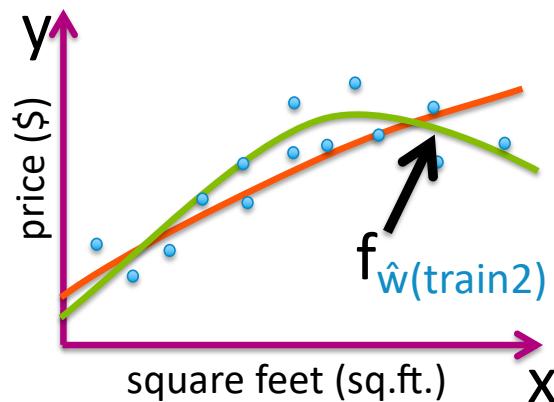
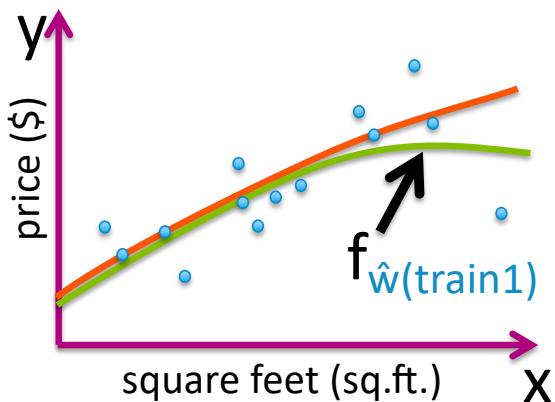


# Bias of function estimator

Average prediction error at  $x_t$

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

Bias how well on avg model can fit  $f_{w(\text{true})}$   
Variance variability of fit



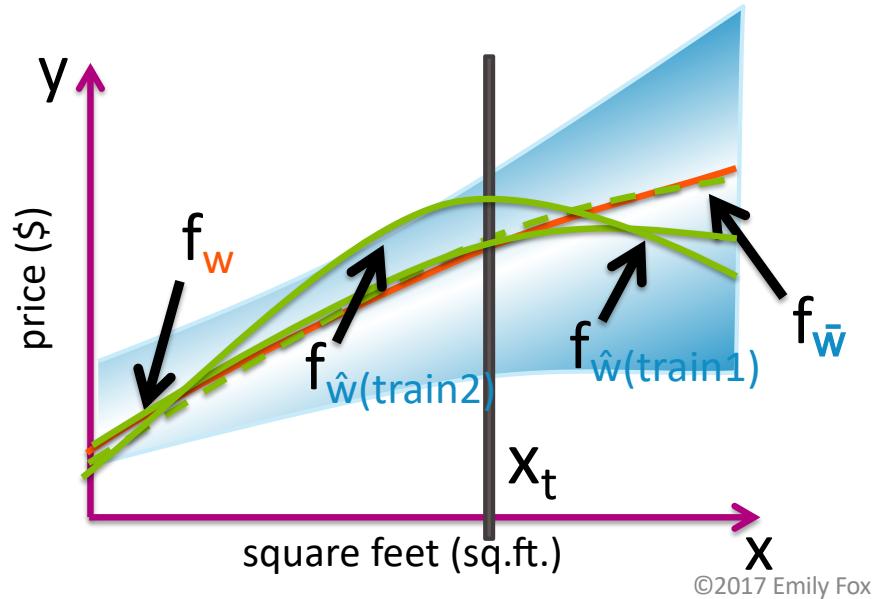
# Bias of function estimator

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

True function =  $f_w(x)$

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

over all training sets of size N

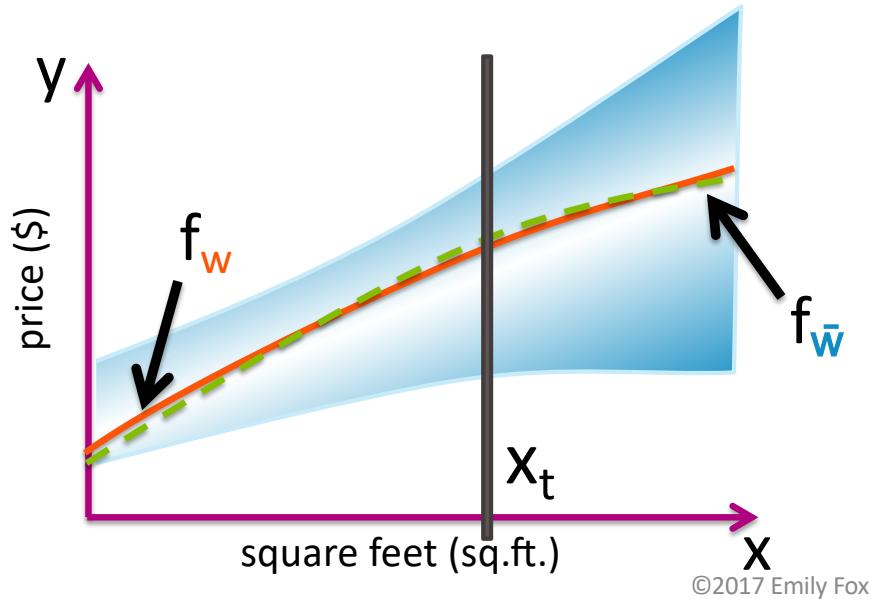


# Bias of function estimator

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  $x_t$

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

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

True function =  $f_w(x)$

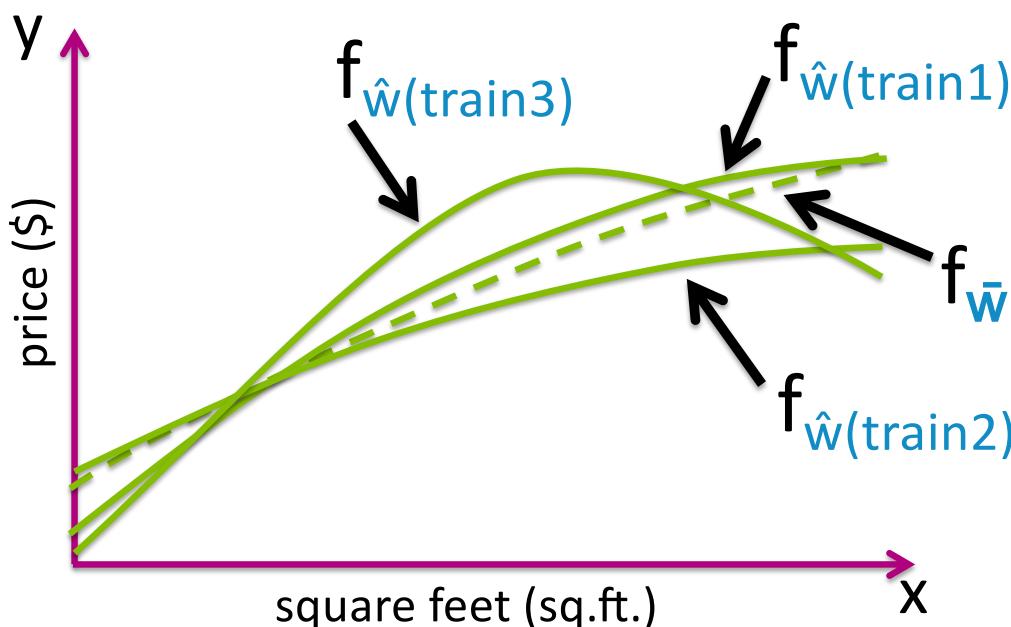
$$\text{bias}(f_{\hat{w}}(x_t)) = f_w(x_t) - f_{\bar{w}}(x_t)$$

# Variance of function estimator

Bias how well on avg model can fit  $f_{\hat{w}(\text{true})}$   
Variance variability of fit

Average prediction error at  $x_t$

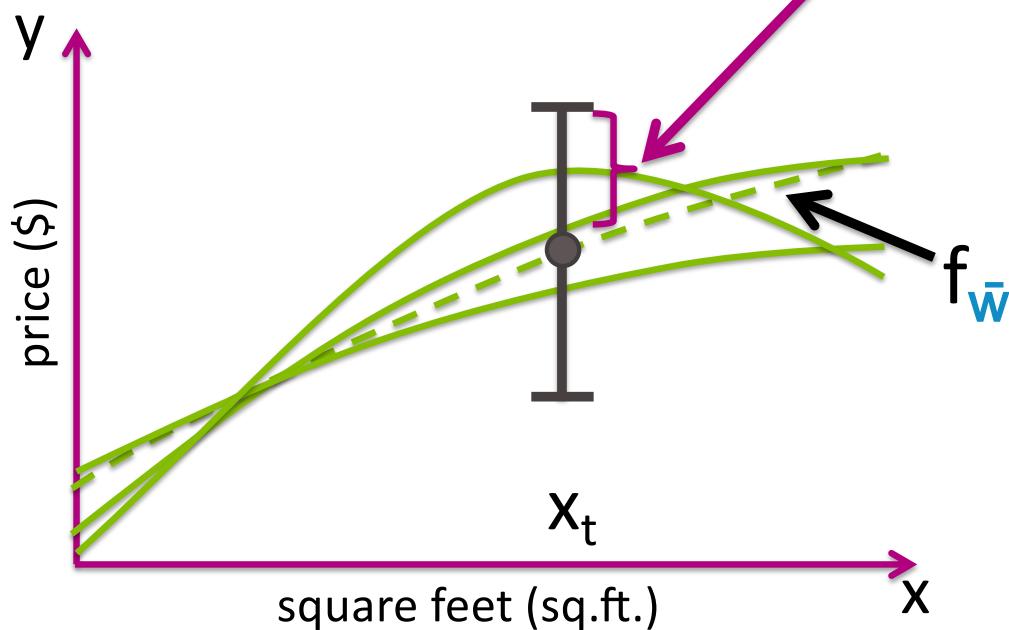
$$= \sigma^2 + [\text{bias}(f_{\hat{w}}(x_t))]^2 + \text{var}(f_{\hat{w}}(x_t))$$



# Variance of function estimator

Average prediction error at  $x_t$

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

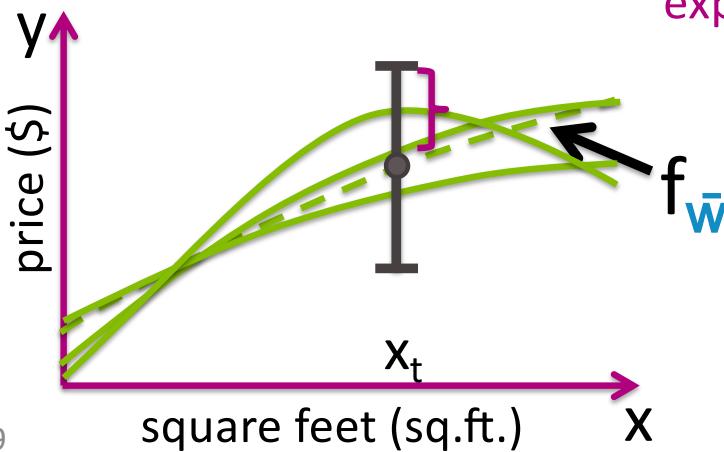


# Variance of function estimator

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

fit on a specific training dataset      what I expect to learn over all training sets average fit

over all training sets of size N      deviation of specific fit from expected fit at  $x_t$



# Sum of 3 sources of error

Average prediction error at  $x_t = E_{y|x}[(y - f_{\hat{w}(\text{train})}(x_t))^2]$

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

$$\sigma^2 \text{ at } x_t = E_{y|x_t}[(y - f_{w(\text{true})}(x_t))^2]$$

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

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

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

# Why 3 sources of error? A formal derivation

# Deriving expected prediction error

Expected prediction error

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

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

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

Expected prediction error at  $x_t$

$$= E_{\text{train}, y|x_t} [(y - f_{\hat{w}(\text{train})}(x_t))^2]$$

# Deriving expected prediction error

Expected prediction error at  $x_t$

$$= E_{\text{train}, y|x_t} [(y - f_{\hat{w}(\text{train})}(x_t))^2]$$

$$= E_{\text{train}, y|x_t} [((y - f_{w(\text{true})}(x_t)) + (f_{w(\text{true})}(x_t) - f_{\hat{w}(\text{train})}(x_t)))^2]$$

# Equating MSE with bias and variance

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

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

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

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

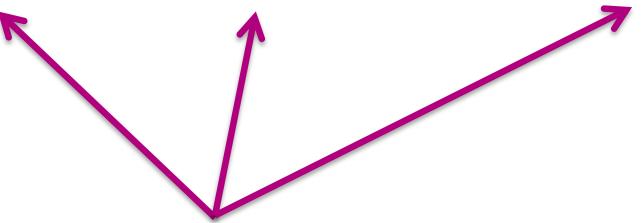
# Putting it all together

Expected prediction error at  $x_t$

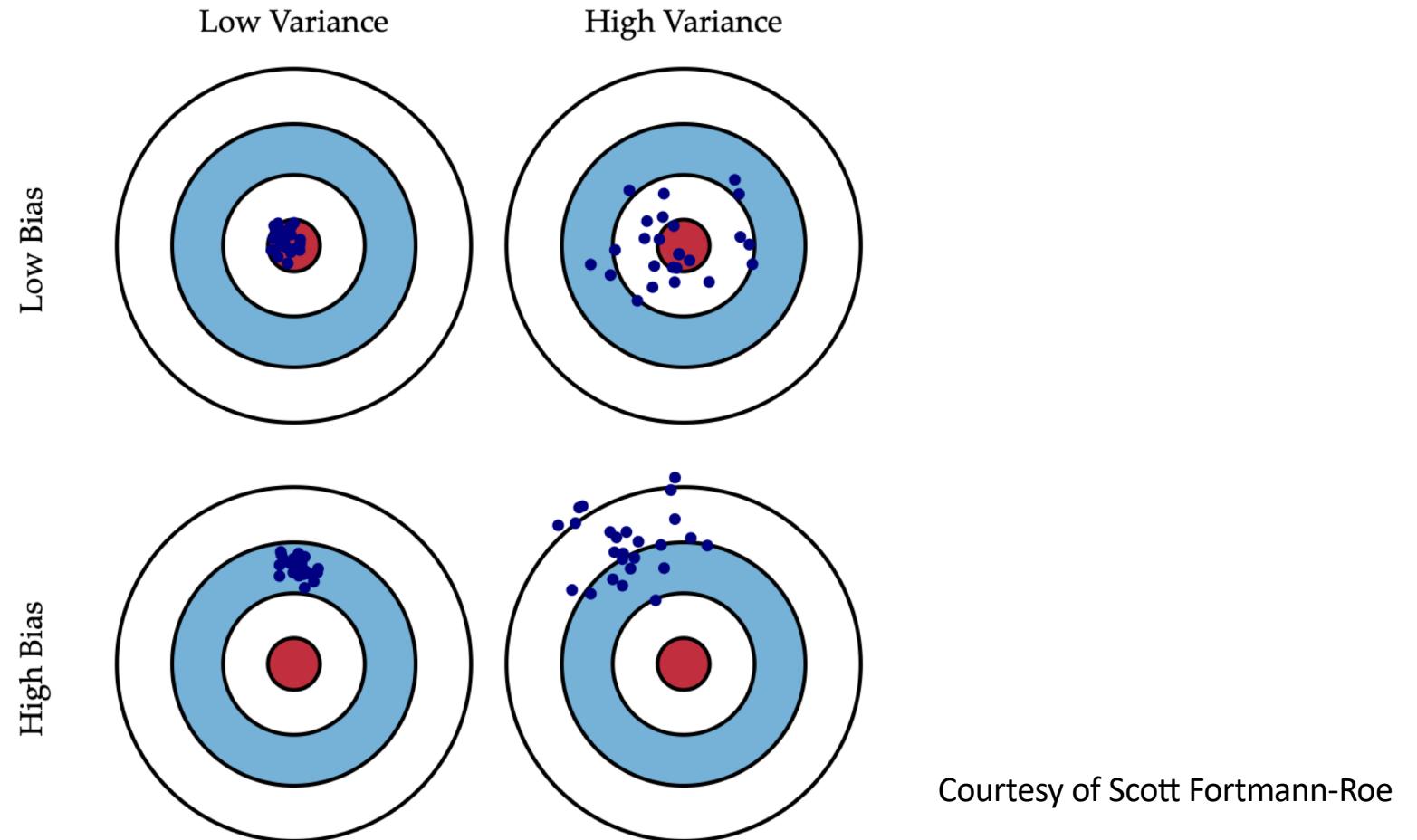
$$= \sigma^2 + \text{MSE}[f_{\hat{w}}(x_t)]$$

$$\text{MSE} = \text{bias}^2 + \text{variance}$$

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



3 sources of error



# 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