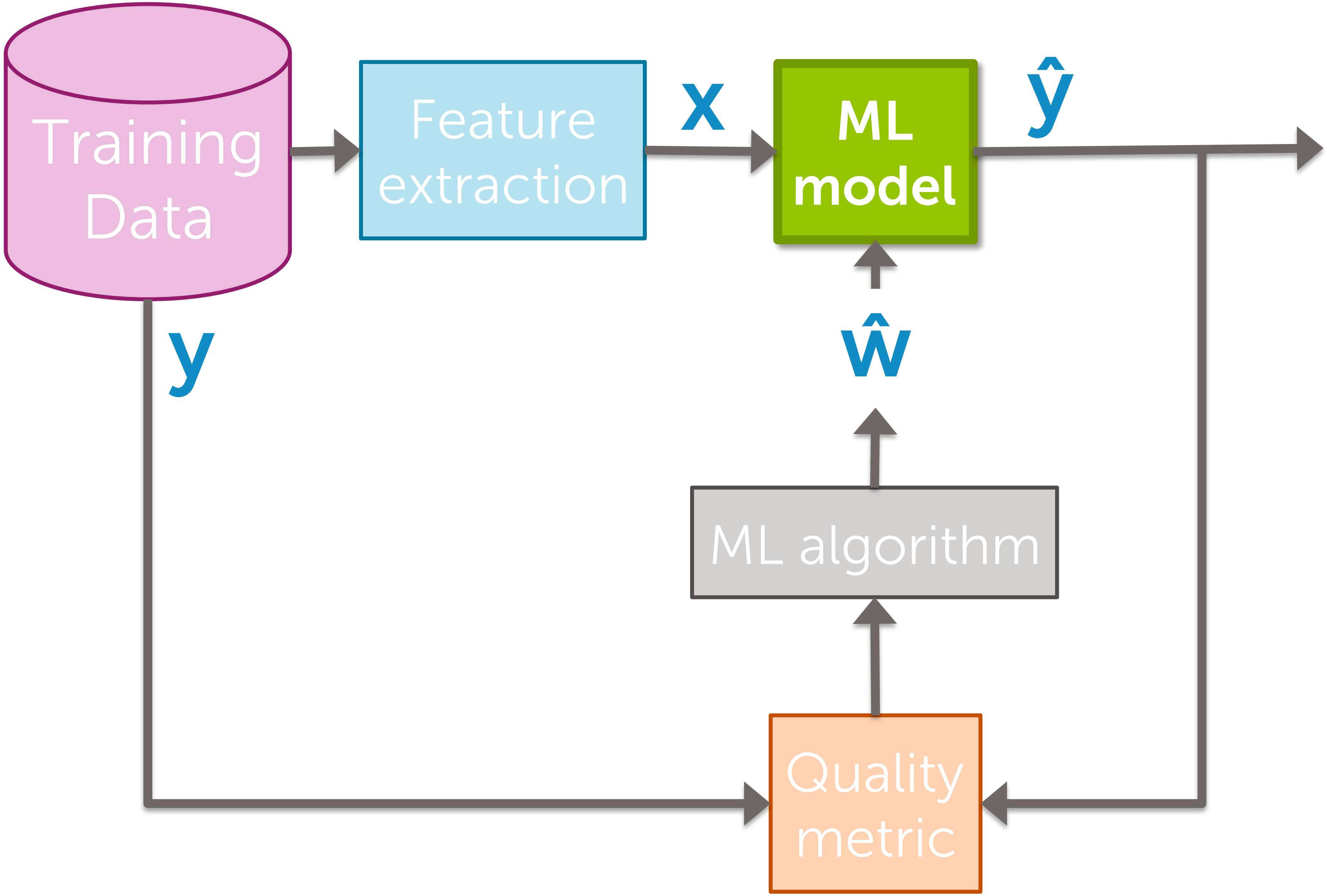
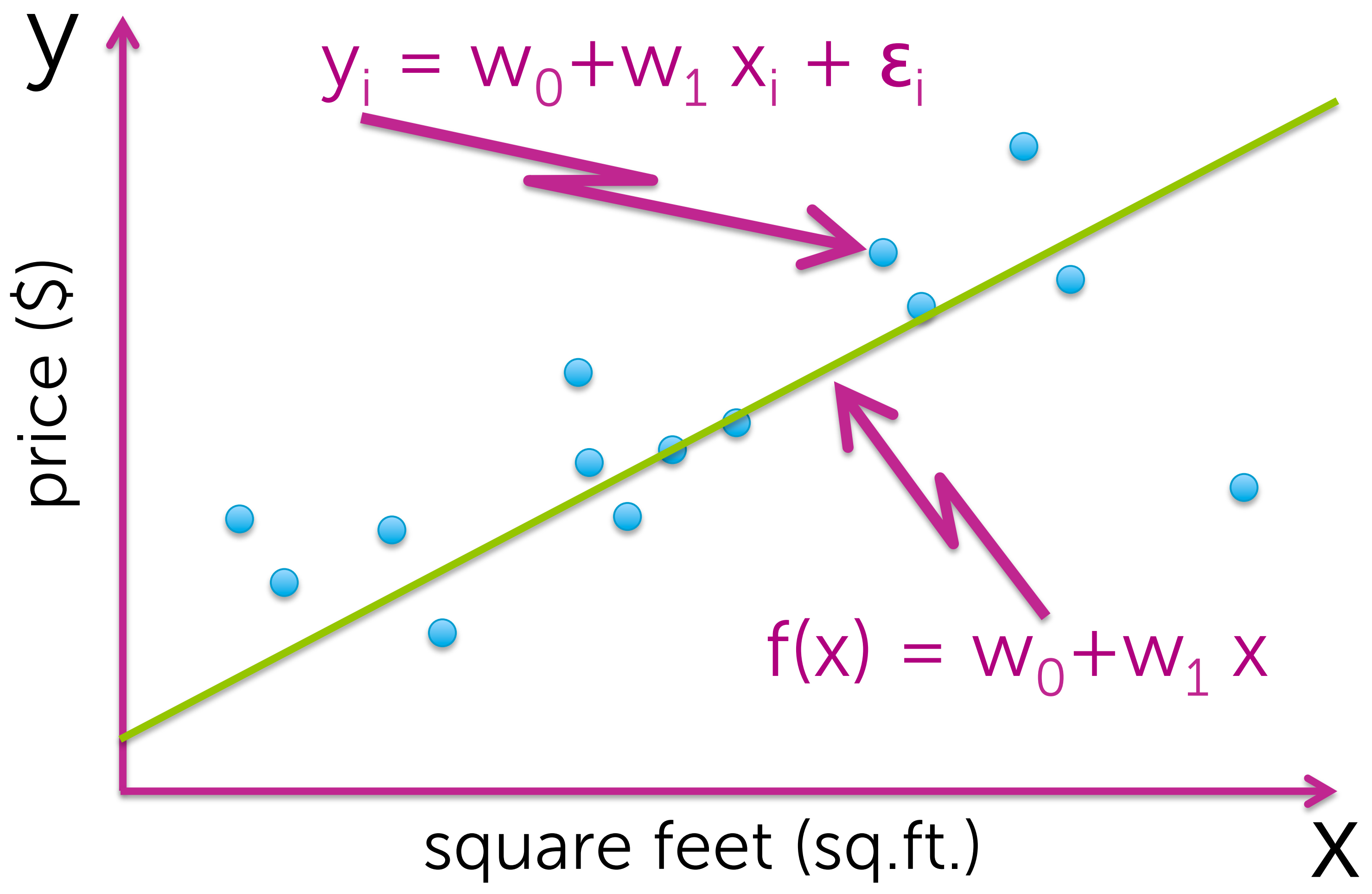
Multiple Regression:

Linear regression with multiple features

Emily Fox & Carlos Guestrin
Machine Learning Specialization
University of Washington


Training Data
Feature extraction
x
ML model
ŷ
y
ŵ
ML algorithm
Quality metric

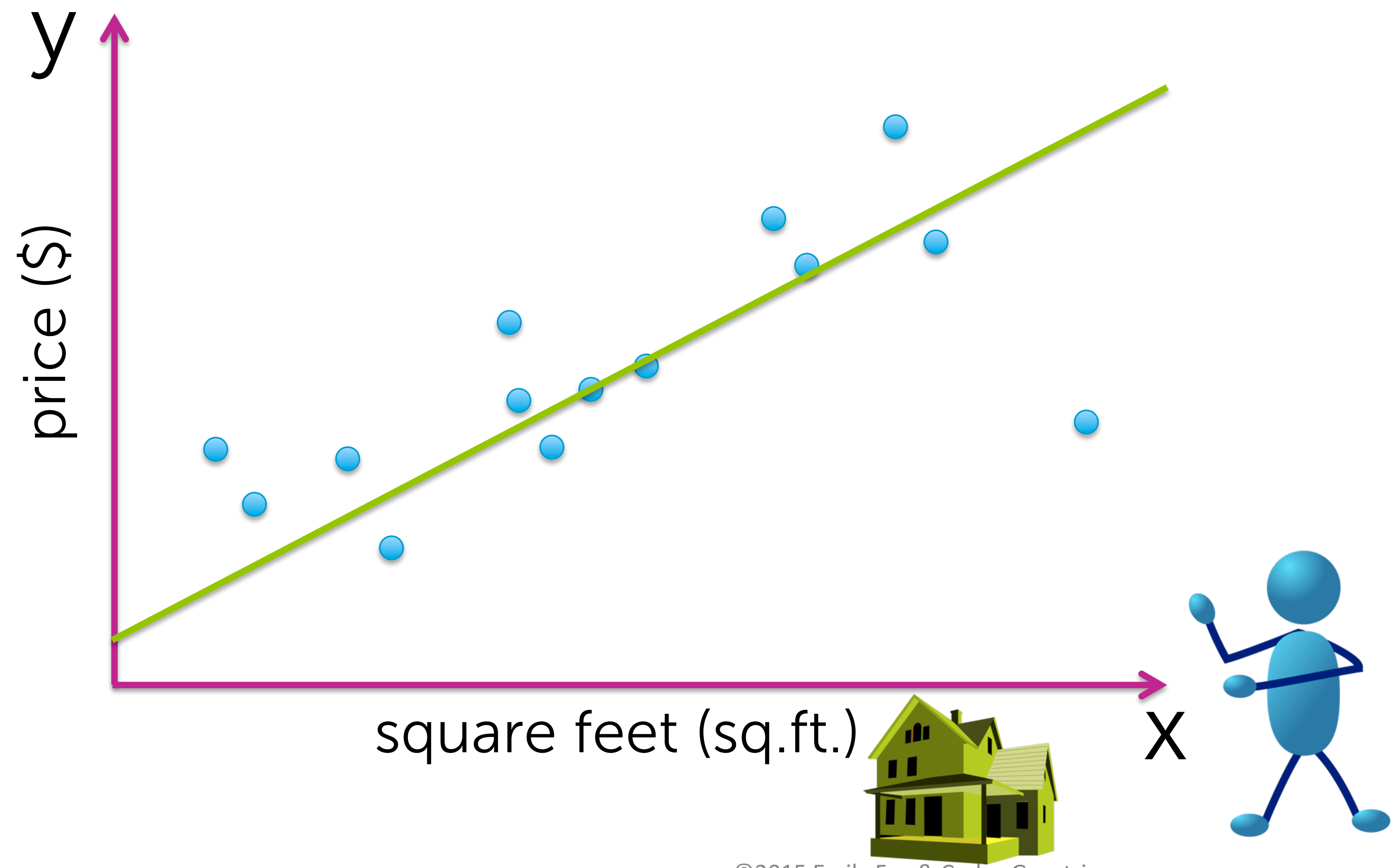
Simple linear regression model



More complex functions of a single input

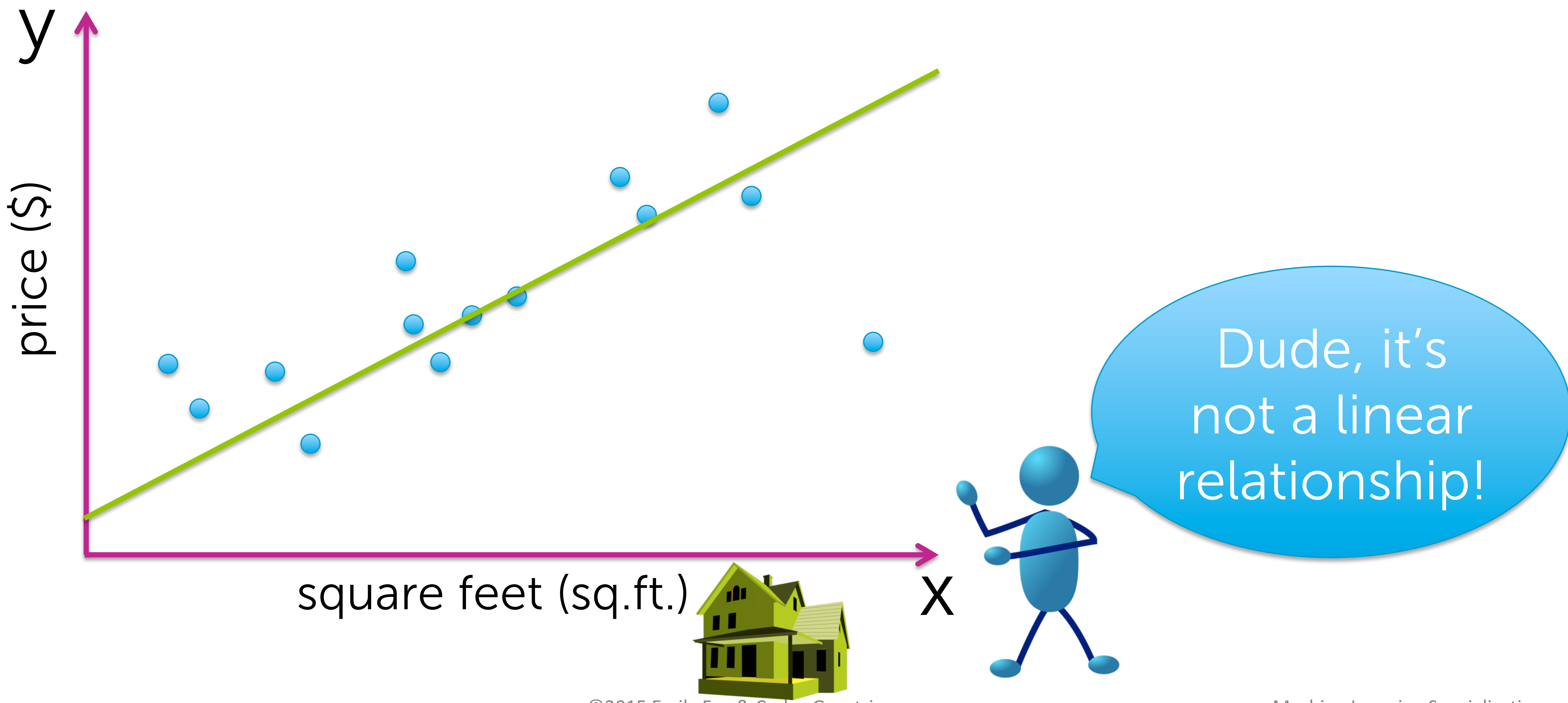
Polynomial regression

Fit data with a line or … ?

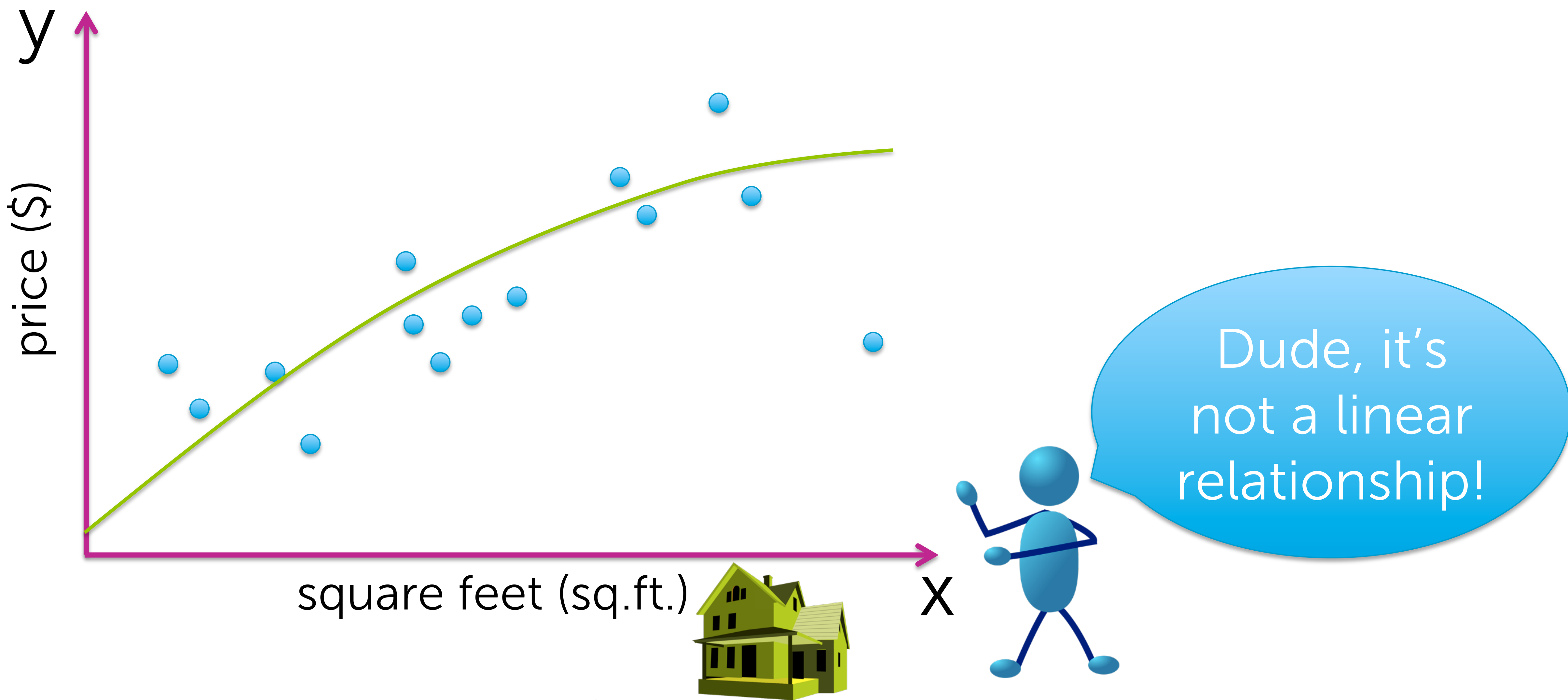


You show your friend your analysis

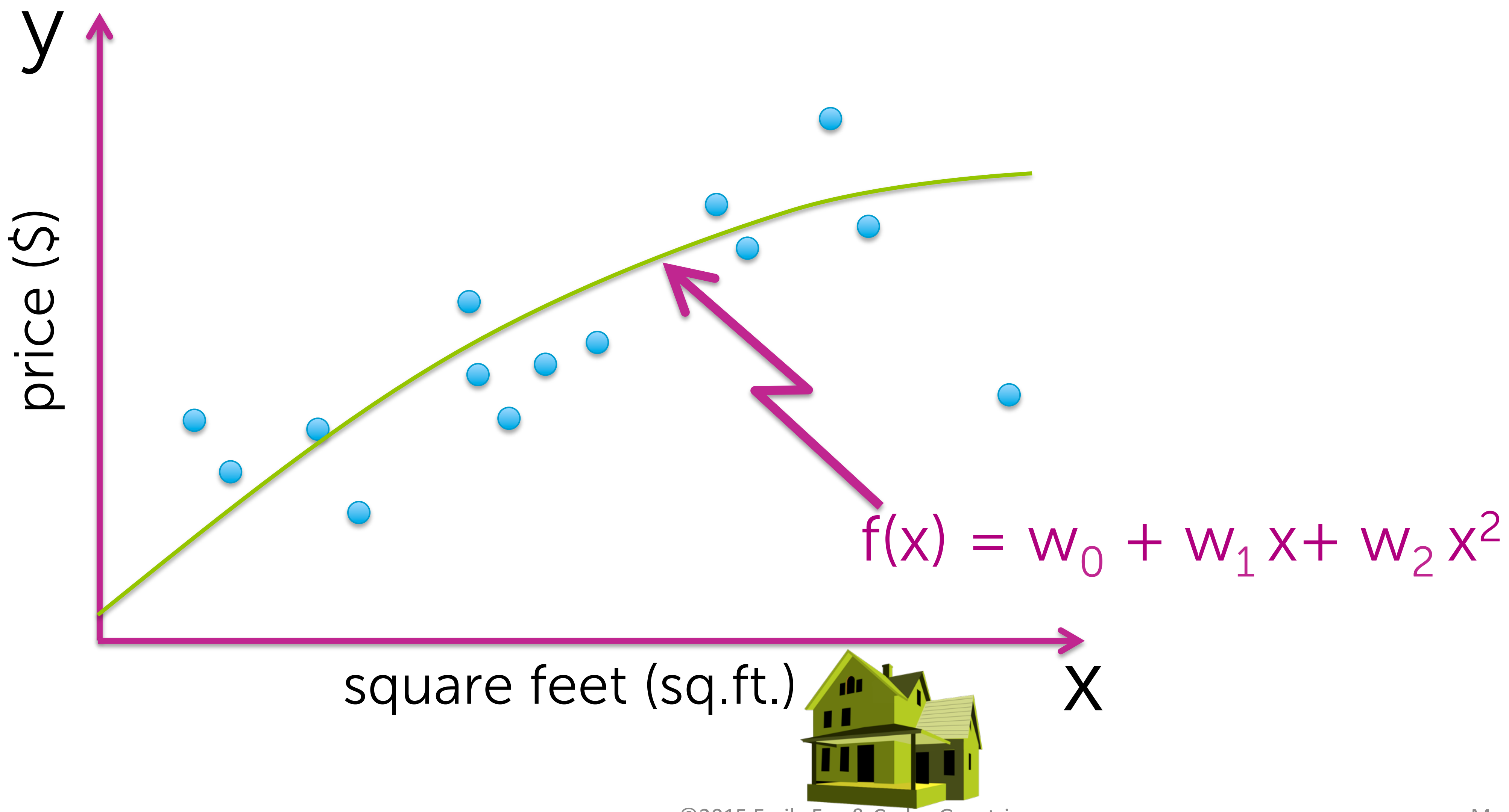
Fit data with a line or … ?



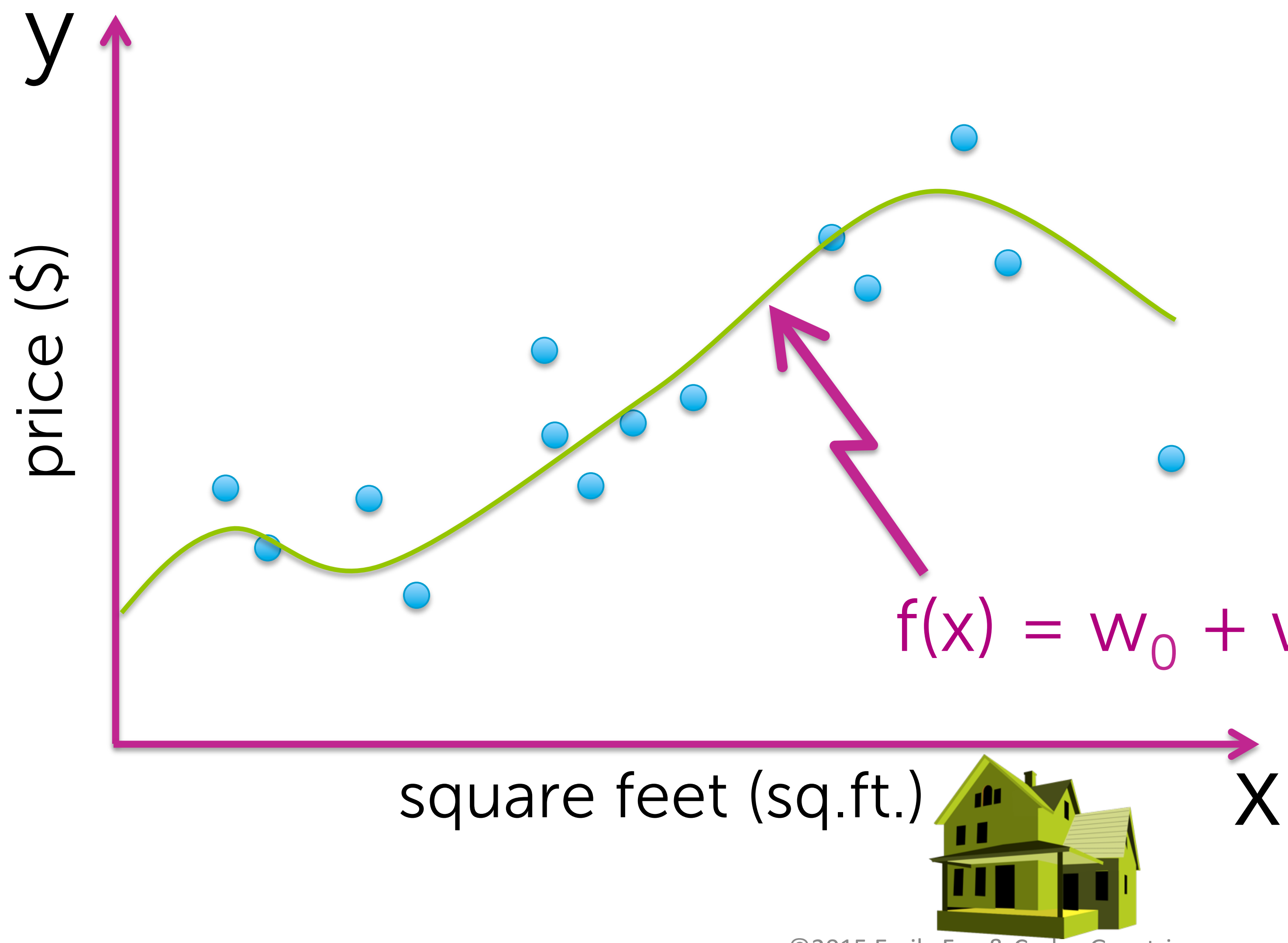
What about a quadratic function?



What about a quadratic function?



Even higher order polynomial



$$f(x) = w_0 + w_1 x + w_2 x^2 + \ldots + w_p x^p$$

Polynomial regression

Model:

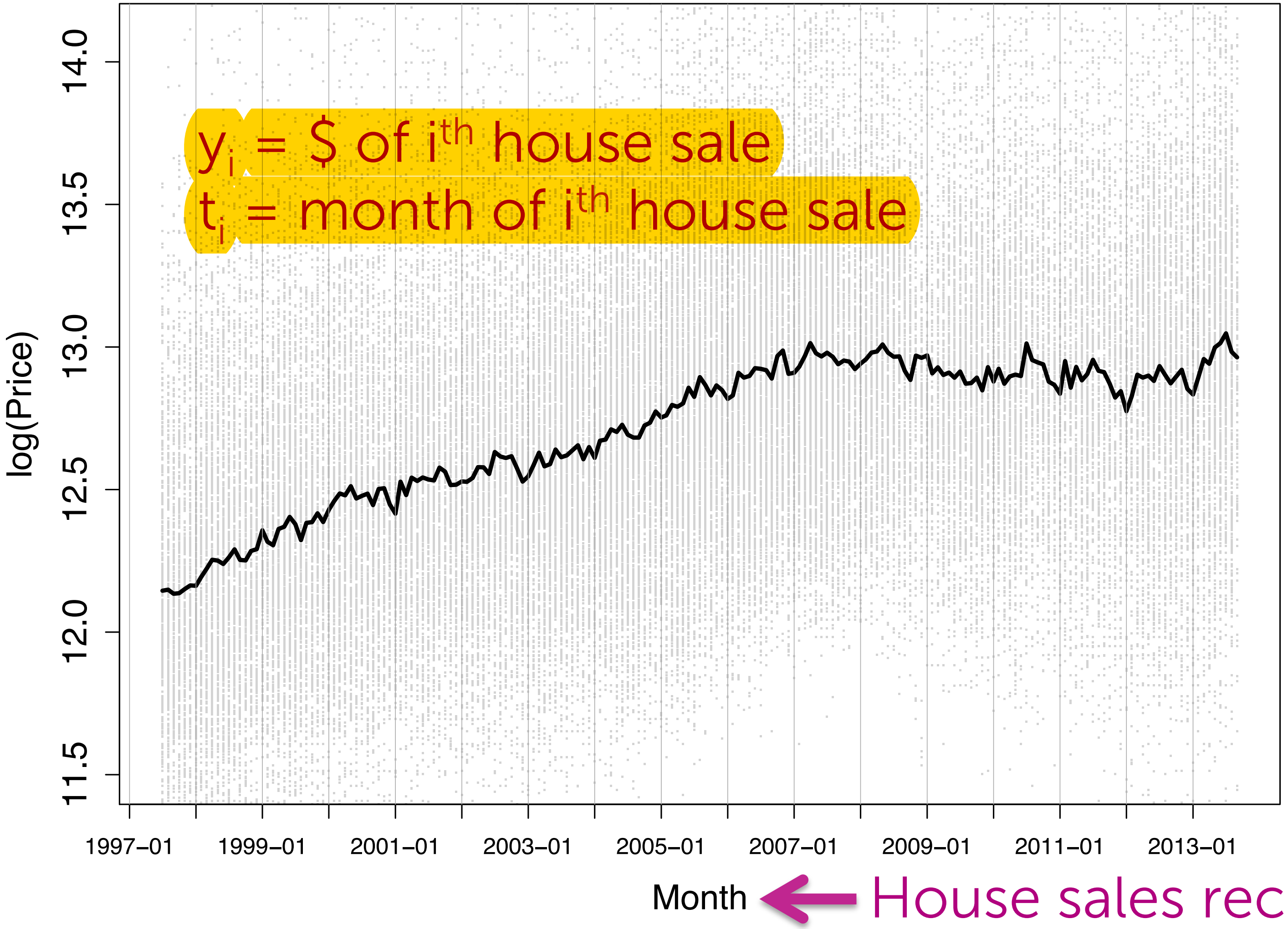
$$y_i = w_0 + w_1 x_i + w_2 x_i^2 + \ldots + w_p x_i^p + \varepsilon_i$$

treat as different **features**

feature 1 = 1 (constant)	*parameter* 1 = w_0
feature 2 = x	*parameter* 2 = w_1
feature 3 = x^2	*parameter* 3 = w_2
…	…
feature p+1 = x^p	*parameter* p+1 = w_p

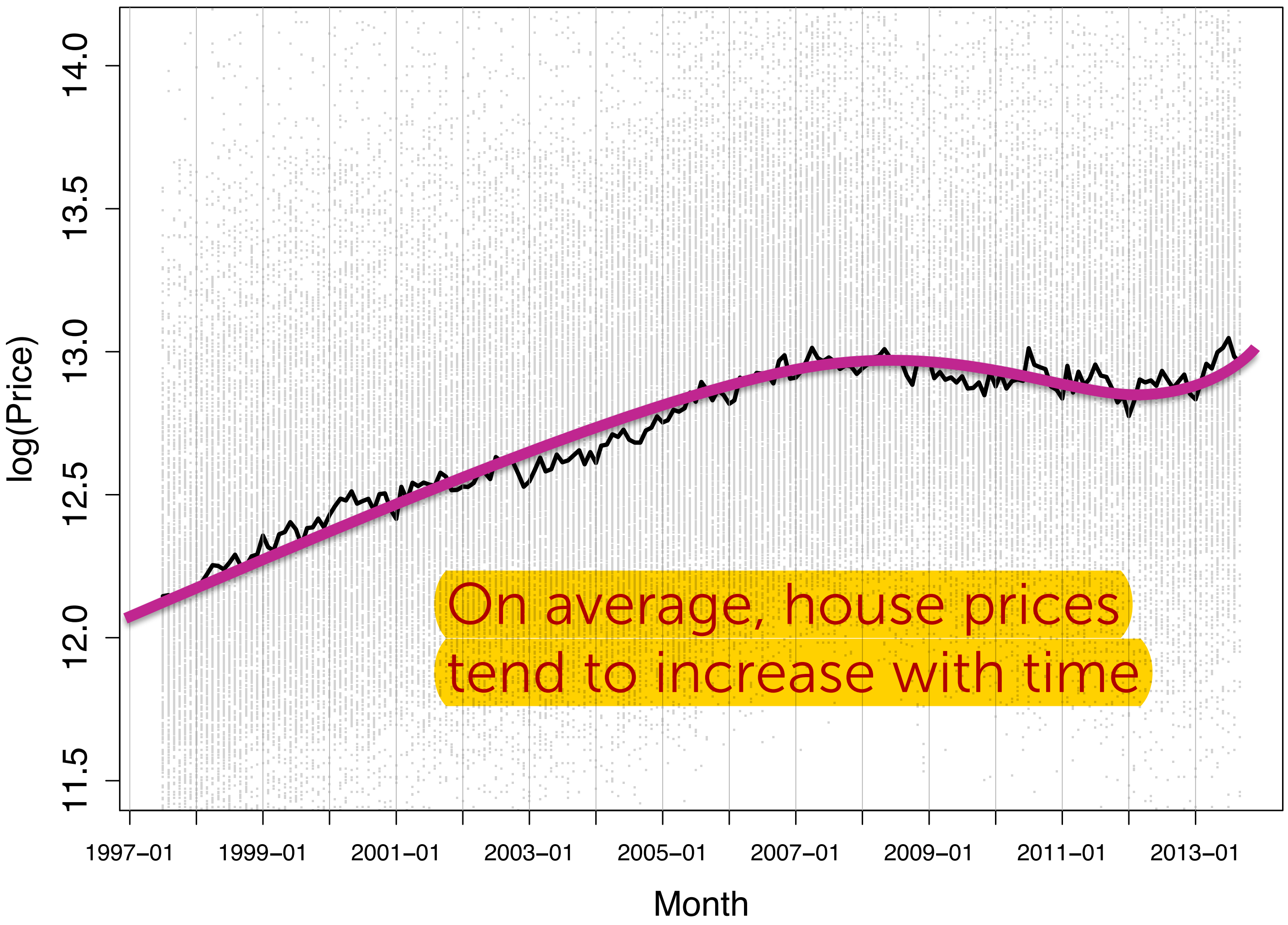
Other functions of one input

Motivating application: Detrending time series

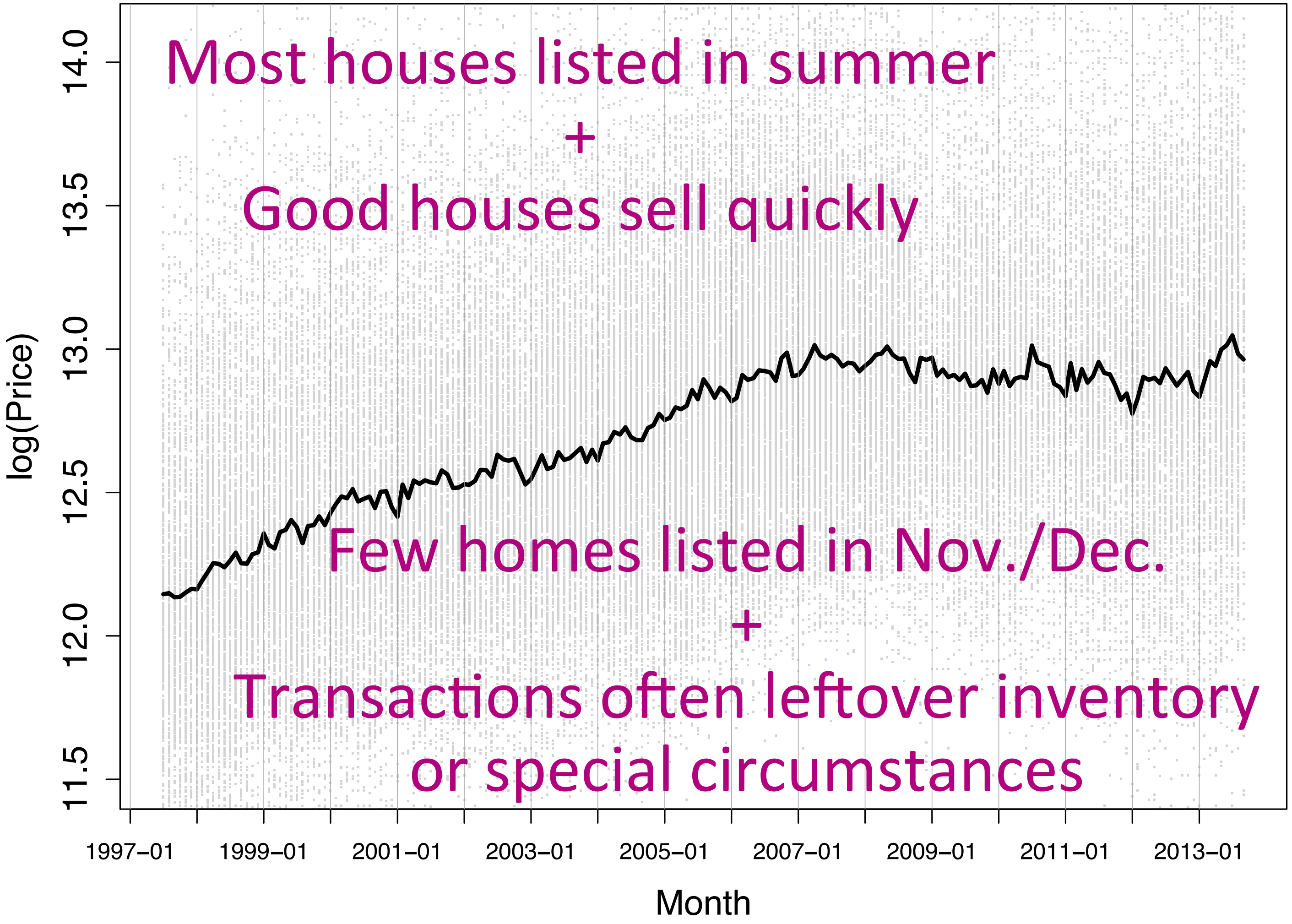


House sales recorded monthly

Trends over time



Seasonality



An example detrending

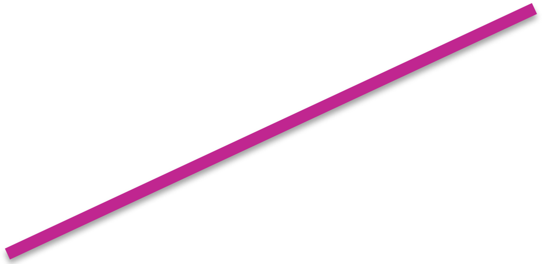
Model:

$$y_i = w_0 + w_1 t_i + w_2 \sin(2\pi t_i / 12 - \Phi) + \varepsilon_i$$

Unknown phase/shift

Linear trend

Seasonal component =
Sinusoid with period 12
(resets annually)



Trigonometric identity: $\sin(a-b)=\sin(a)\cos(b)-\cos(a)\sin(b)$

$\rightarrow \sin(2\pi t_i / 12 - \Phi) = \sin(2\pi t_i / 12)\cos(\Phi) - \cos(2\pi t_i / 12)\sin(\Phi)$

An example detrending

Equivalently,

$$y_i = w_0 + w_1 t_i + w_2 \sin(2\pi t_i / 12) + w_3 \cos(2\pi t_i / 12) + \varepsilon_i$$

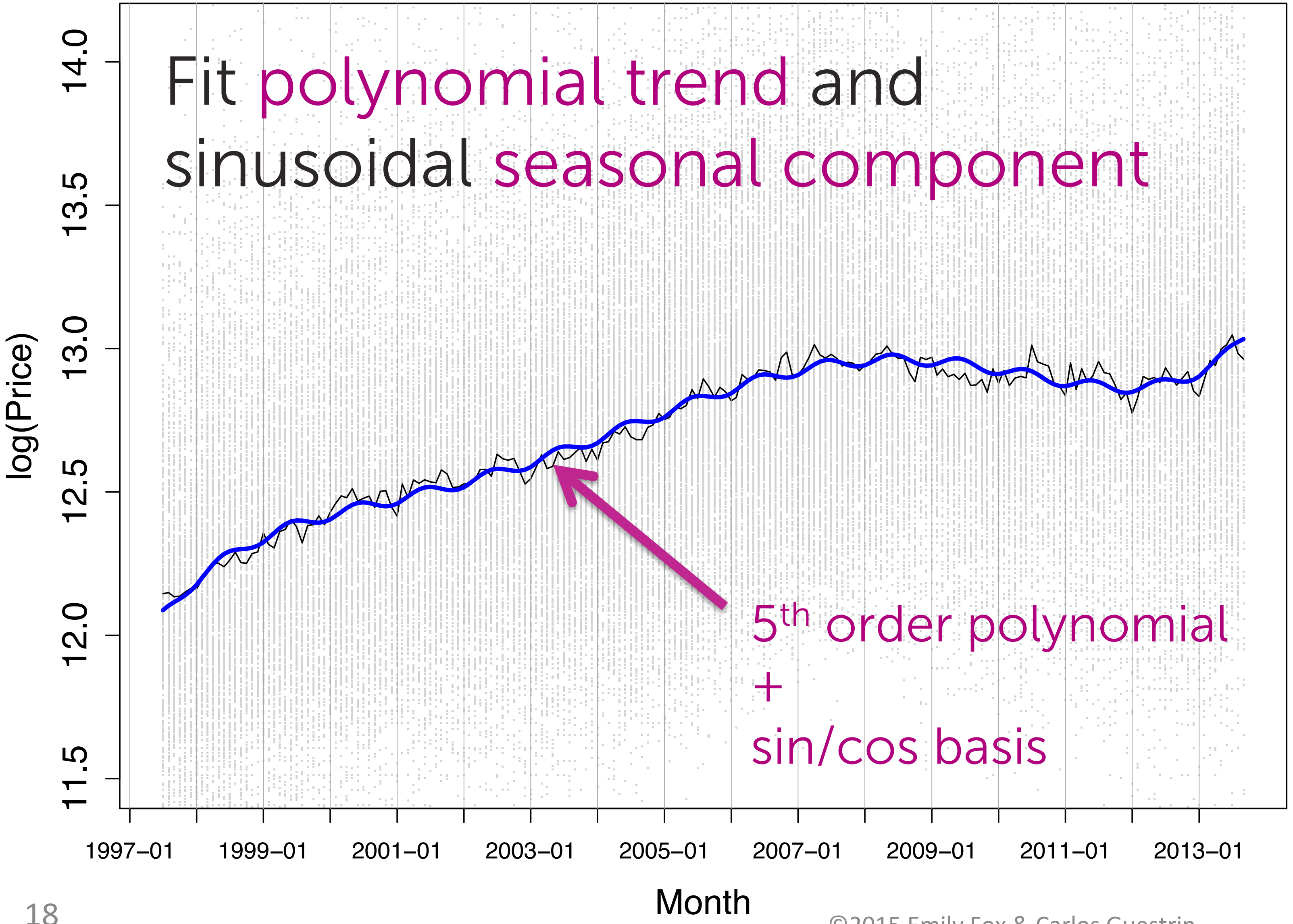
feature 1 = 1 (constant)

feature 2 = t

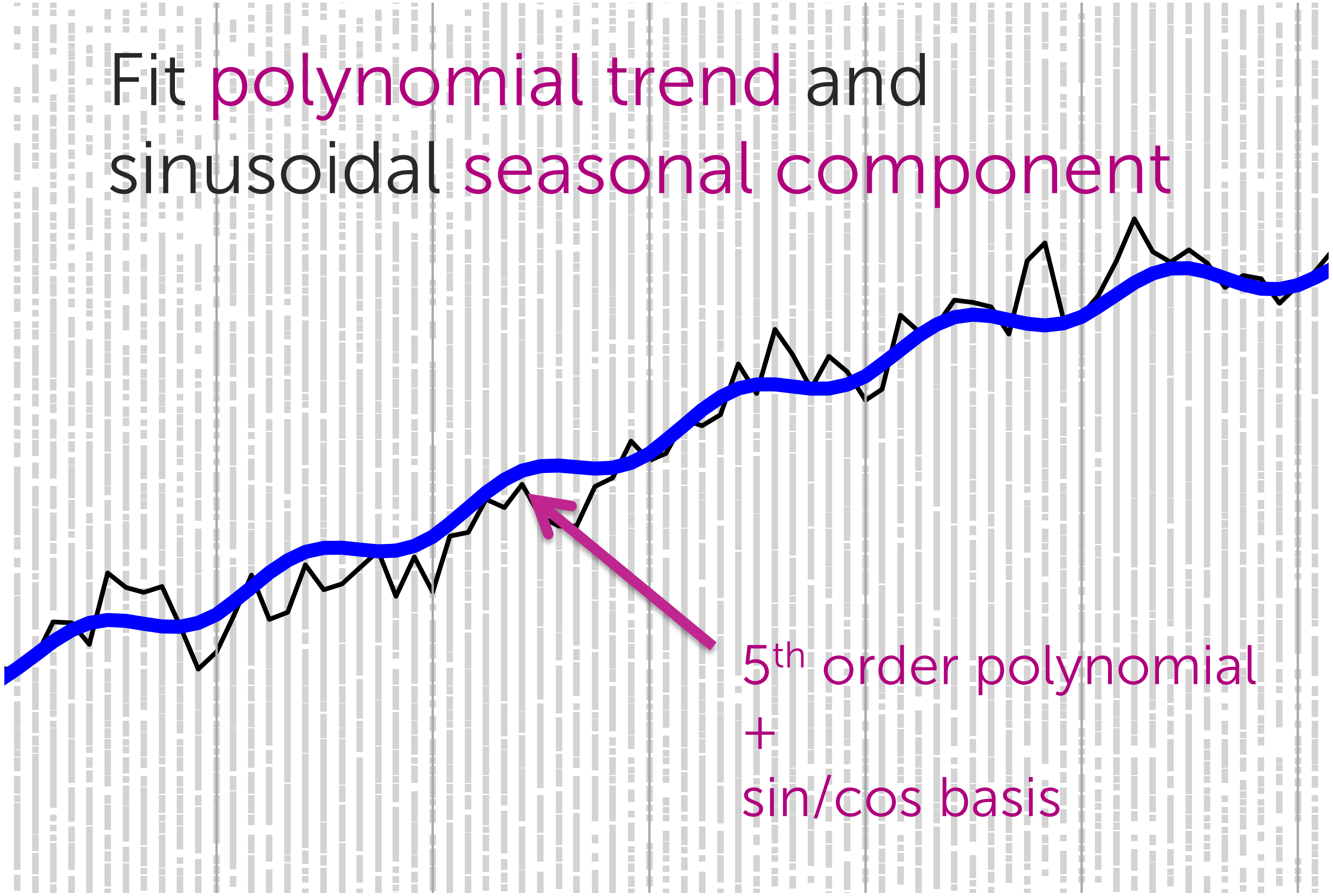
feature 3 = sin(2πt/12)

feature 4 = cos(2πt/12)

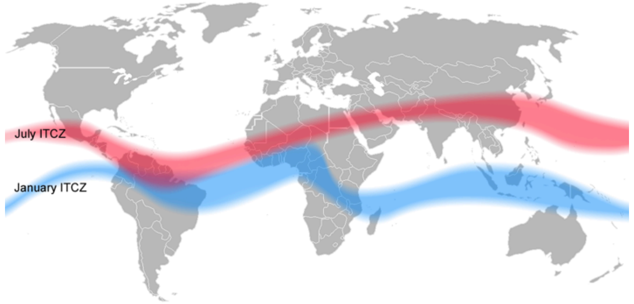
Detrended housing data



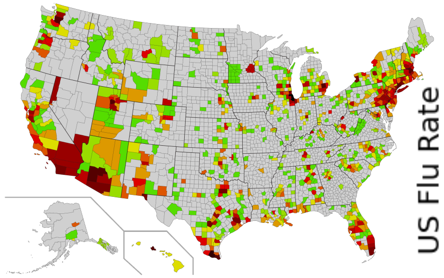
Zoom in…


Fit polynomial trend and
sinusoidal seasonal component
5th order polynomial
+
sin/cos basis

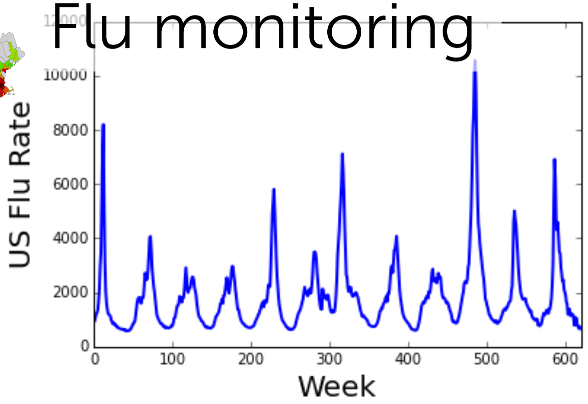
Other examples of seasonality



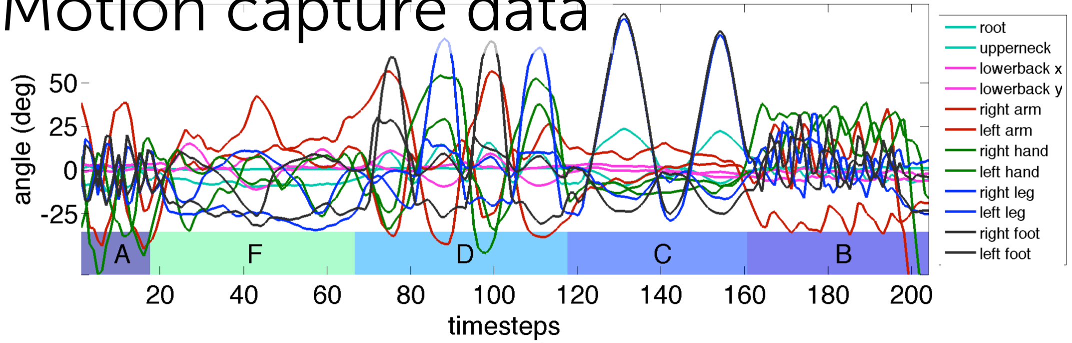
Weather modeling
(e.g., temperature, rainfall)



Flu monitoring



Motion capture data



More generally…

Generic basis expansion

Model:

$$y_i = w_0 h_0(x_i) + w_1 h_1(x_i) + \ldots + w_D h_D(x_i) + \varepsilon_i$$

$$= \sum_{j=0}^{D} w_j h_j(x_i) + \varepsilon_i$$

j^{th} *feature*

j^{th} *regression coefficient* or *weight*

Generic basis expansion

Model:

$$y_i = w_0 h_0(x_i) + w_1 h_1(x_i) + \ldots + w_D h_D(x_i) + \varepsilon_i$$

$$= \sum_{j=0}^{D} w_j h_j(x_i) + \varepsilon_i$$

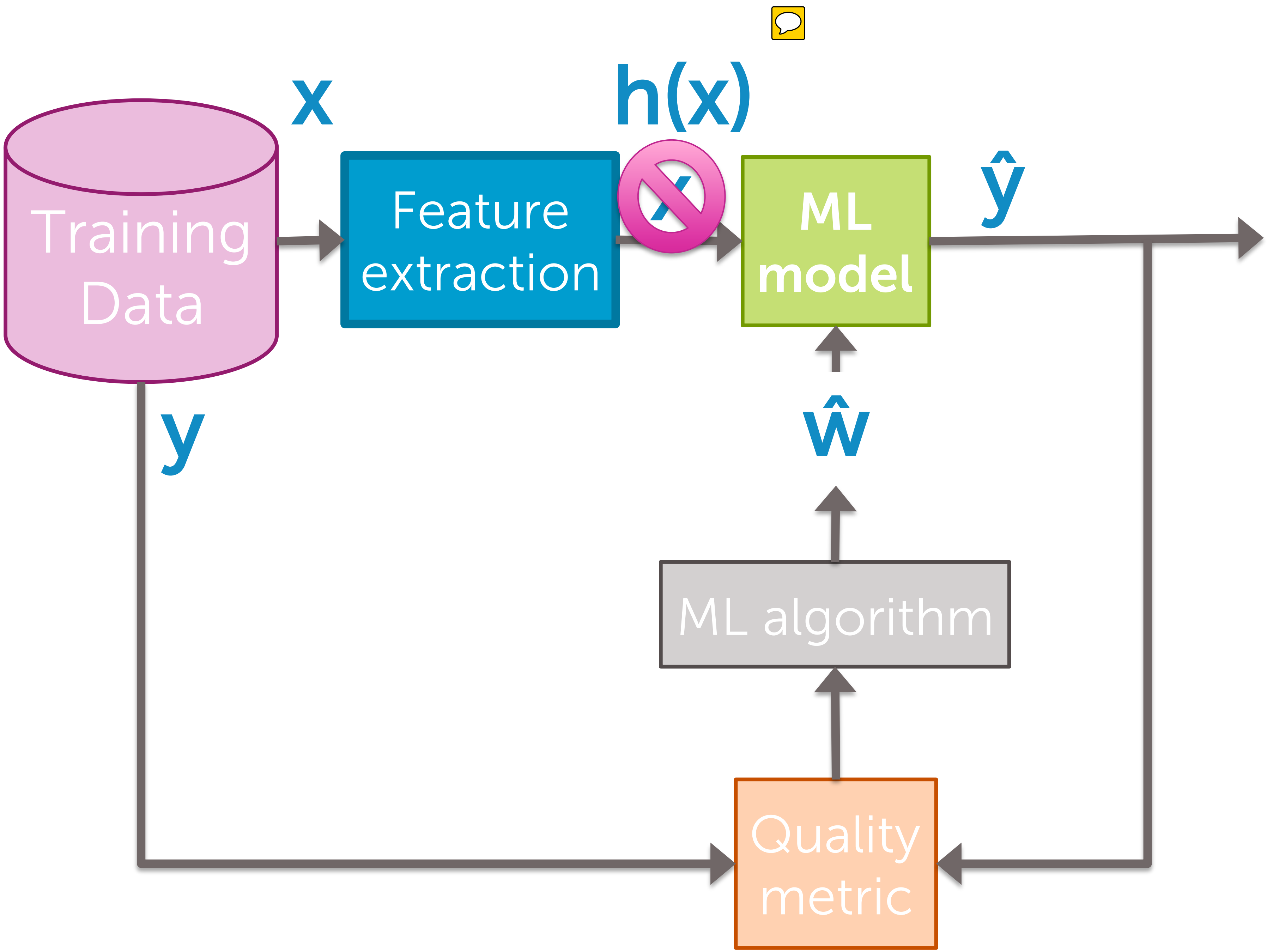
feature 1 = $h_0(x)$…often 1 (constant)

feature 2 = $h_1(x)$… e.g., x

feature 3 = $h_2(x)$… e.g., x^2 or $\sin(2\pi x/12)$

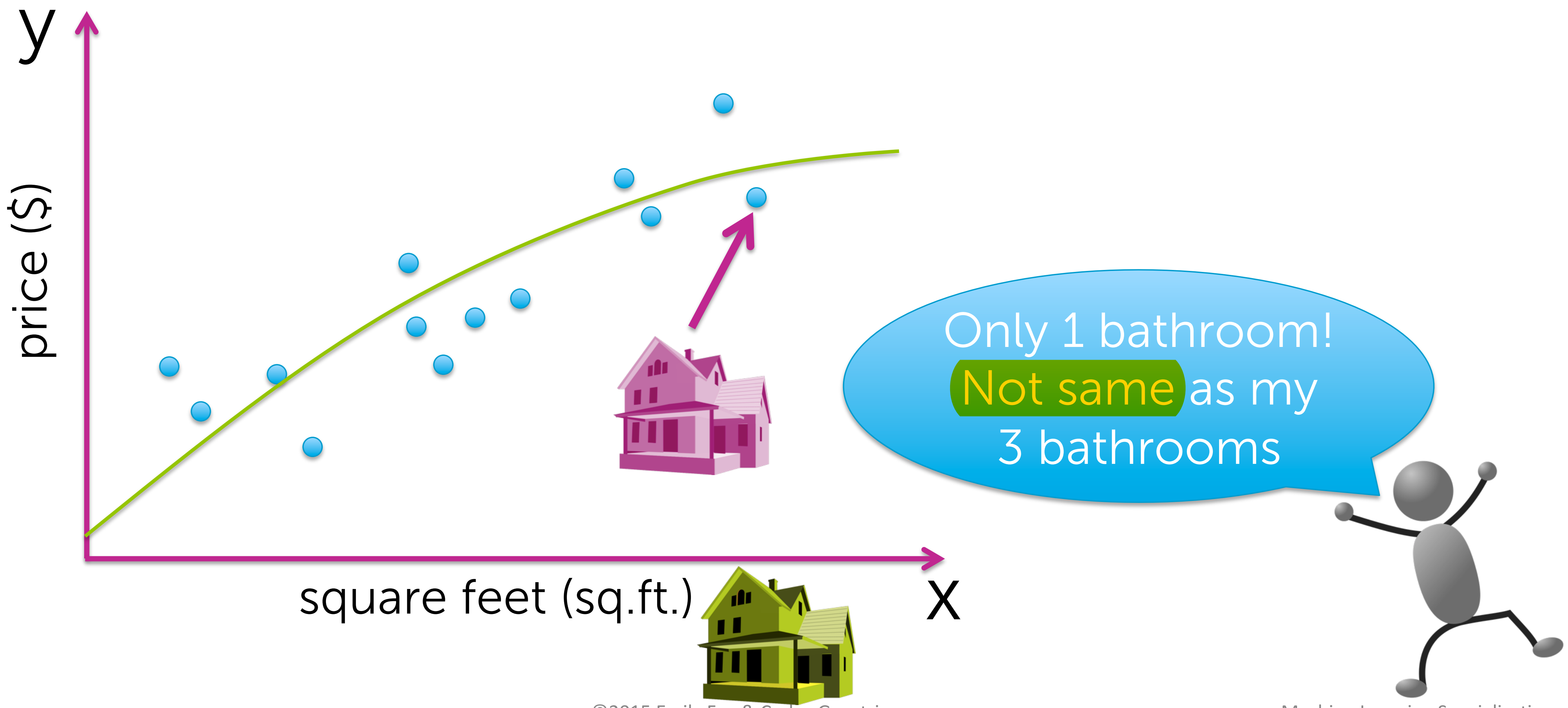
…

feature D+1 = $h_D(x)$… e.g., x^p


x
h(x)
Training Data
Feature extraction
x
ML model
ŷ
y
ŵ
ML algorithm
Quality metric

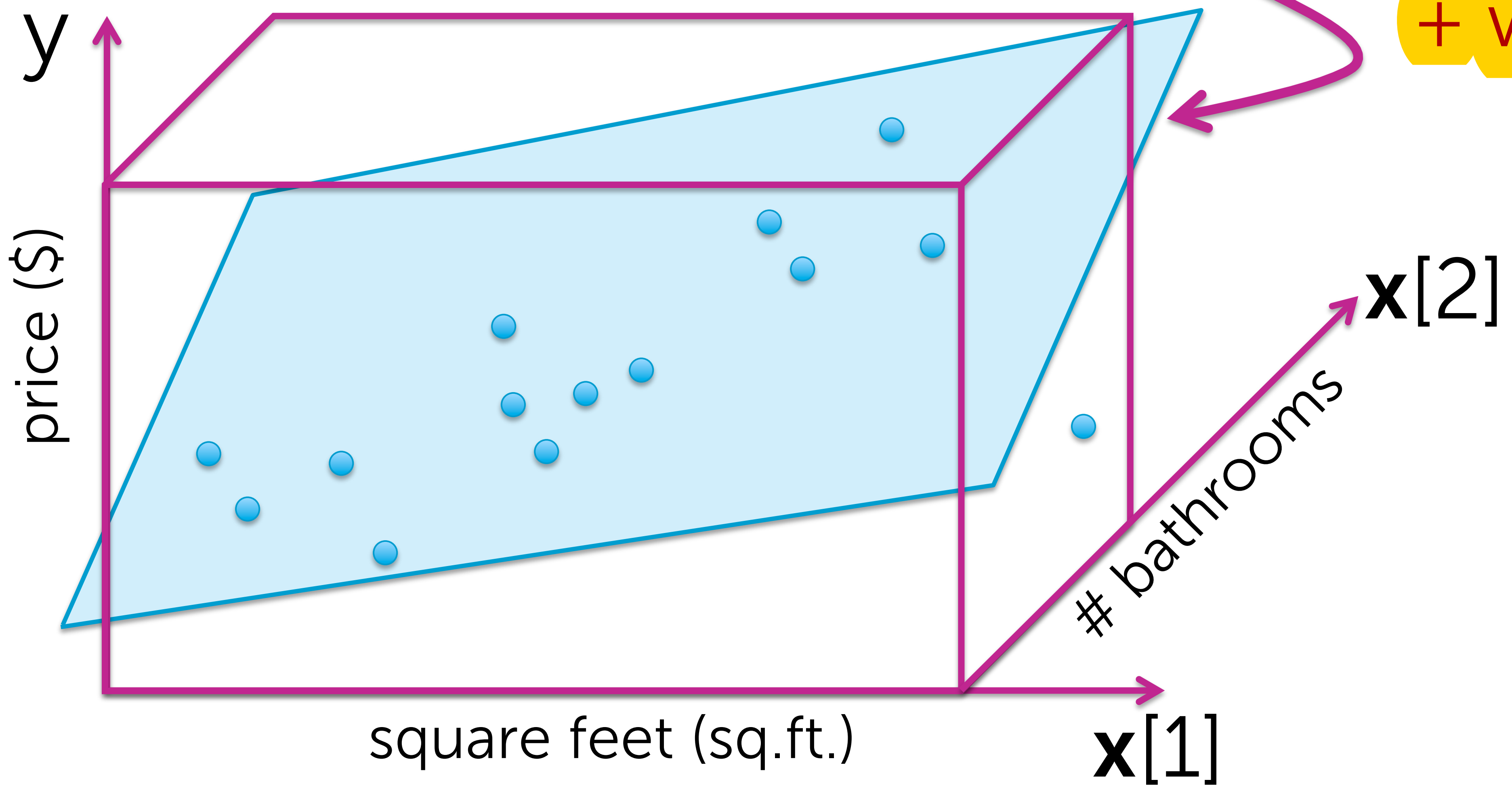
Incorporating multiple inputs

Predictions just based on house size



Add more inputs

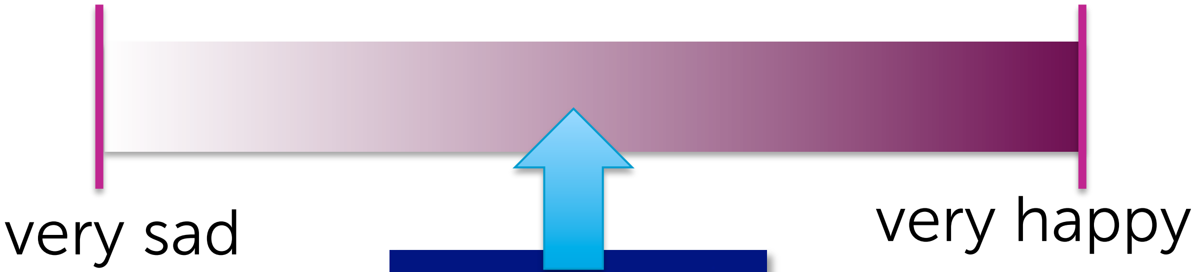
$f(\mathbf{x}) = w_0 + w_1 \text{ sq.ft.} + w_2 \text{ \#bath}$

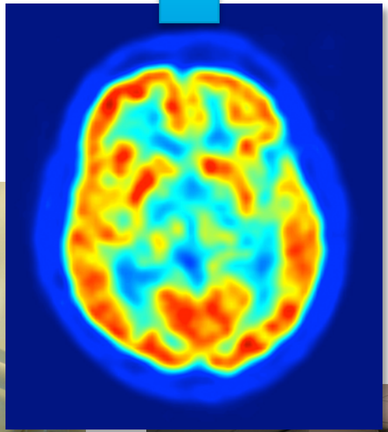


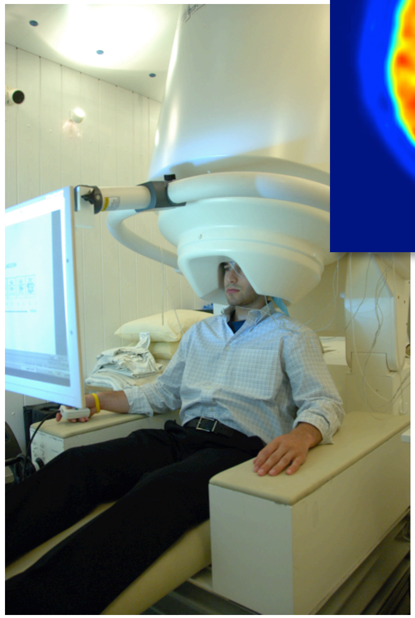
Many possible inputs

- Square feet
- # bathrooms
- # bedrooms
- Lot size
- Year built
- …

Reading your mind







Features are brain region intensities

General notation

Output: y ← scalar

Inputs: $\mathbf{x}$ = ($\mathbf{x}$[1],$\mathbf{x}$[2],…, $\mathbf{x}$[d])

↑ d-dim vector

Notational conventions:

- $\mathbf{x}[j]$ = j^{th} input (*scalar*)
- $h_j(\mathbf{x})$ = j^{th} feature (*scalar*)
- $\mathbf{x}_i$ = input of i^{th} data point (*vector*)
- $\mathbf{x}_i[j]$ = j^{th} input of i^{th} data point (*scalar*)

Simple hyperplane

Model:

$y_i = w_0 + w_1\, \mathbf{x}_i[1] + \ldots + w_d\, \mathbf{x}_i[d] + \varepsilon_i$

feature 1 = 1

feature 2 = $\mathbf{x}[1]$ … e.g., sq. ft.

feature 3 = $\mathbf{x}[2]$ … e.g., #bath

…

feature d+1 = $\mathbf{x}[d]$ … e.g., lot size

More generically…

D-dimensional curve

Model:

$$y_i = w_0\, h_0(\mathbf{x}_i) + w_1\, h_1(\mathbf{x}_i) + \ldots + w_D\, h_D(\mathbf{x}_i) + \varepsilon_i$$

$$= \sum_{j=0}^{D} w_j\, h_j(\mathbf{x}_i) + \varepsilon_i$$

feature 1 = $h_0(\mathbf{x})$ … e.g., 1

feature 2 = $h_1(\mathbf{x})$ … e.g., $\mathbf{x}[1]$ = sq. ft.

feature 3 = $h_2(\mathbf{x})$ … e.g., $\mathbf{x}[2]$ = #bath
or, $\log(\mathbf{x}[7])\, \mathbf{x}[2]$ = log(#bed) x #bath

…

feature D+1 = $h_D(\mathbf{x})$ … some other function of $\mathbf{x}[1]$,…, $\mathbf{x}[d]$

More on notation

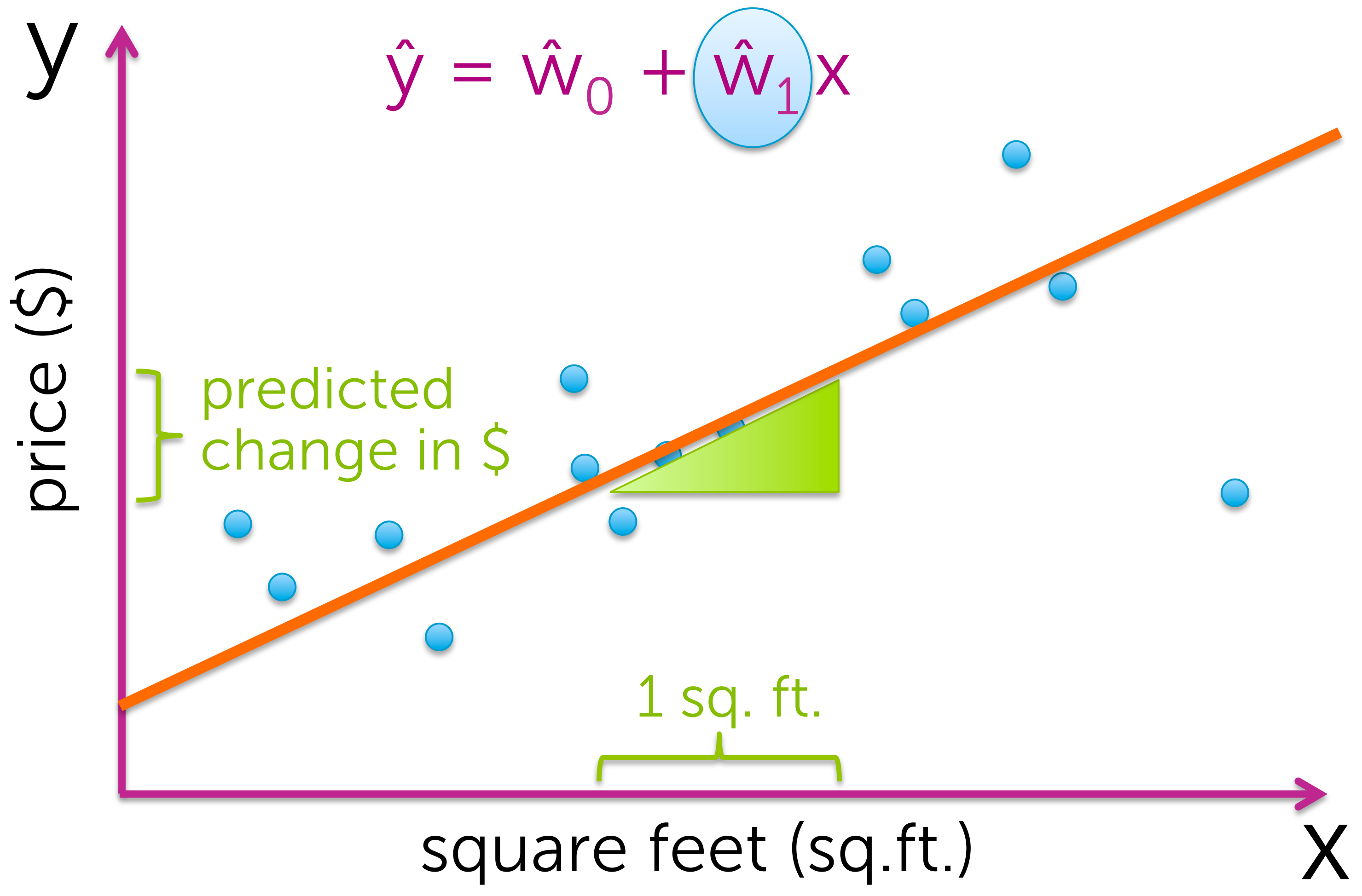
\# observations $(\mathbf{x}_i, y_i)$: N

\# inputs $\mathbf{x}[j]$: d

\# features $h_j(\mathbf{x})$: D

Interpreting the fitted function

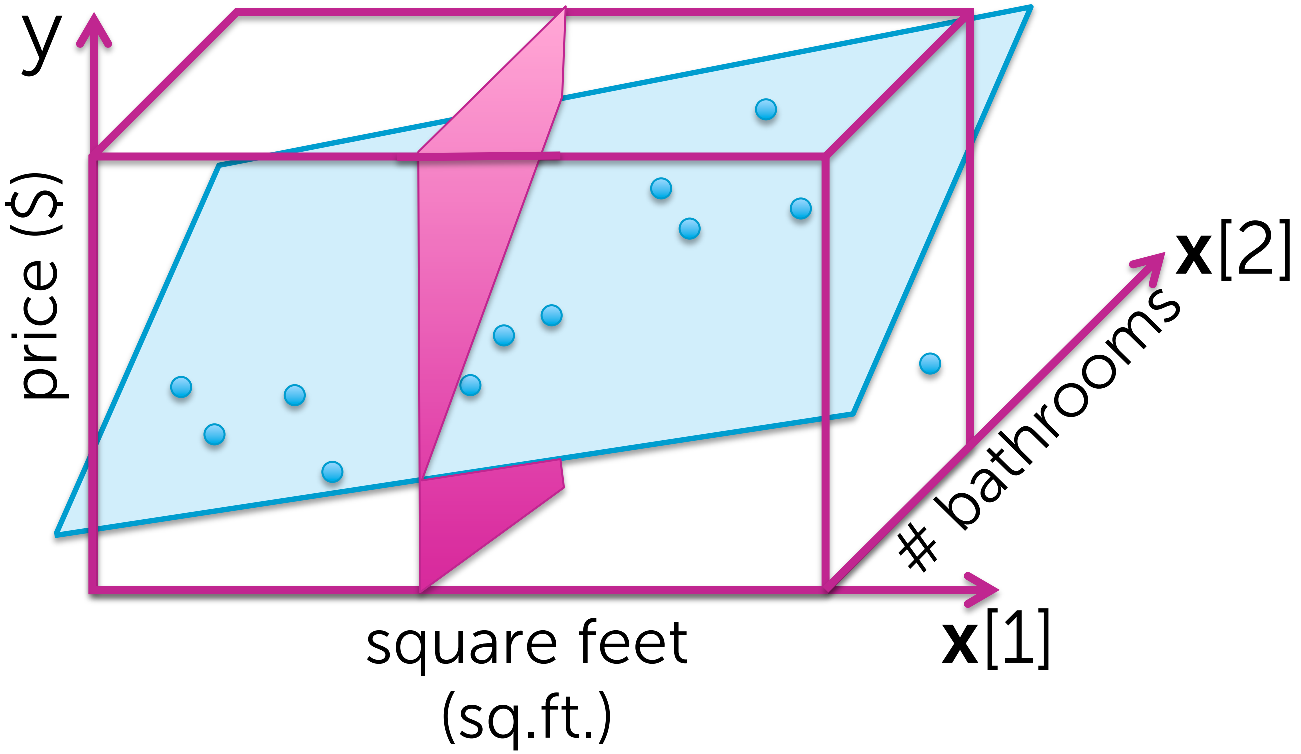
Interpreting the coefficients – Simple linear regression



Interpreting the coefficients – Two linear features

$$\hat{y} = \hat{w}_0 + \hat{w}_1 \boxed{\mathbf{x}[1]} + \hat{w}_2\, \mathbf{x}[2]$$

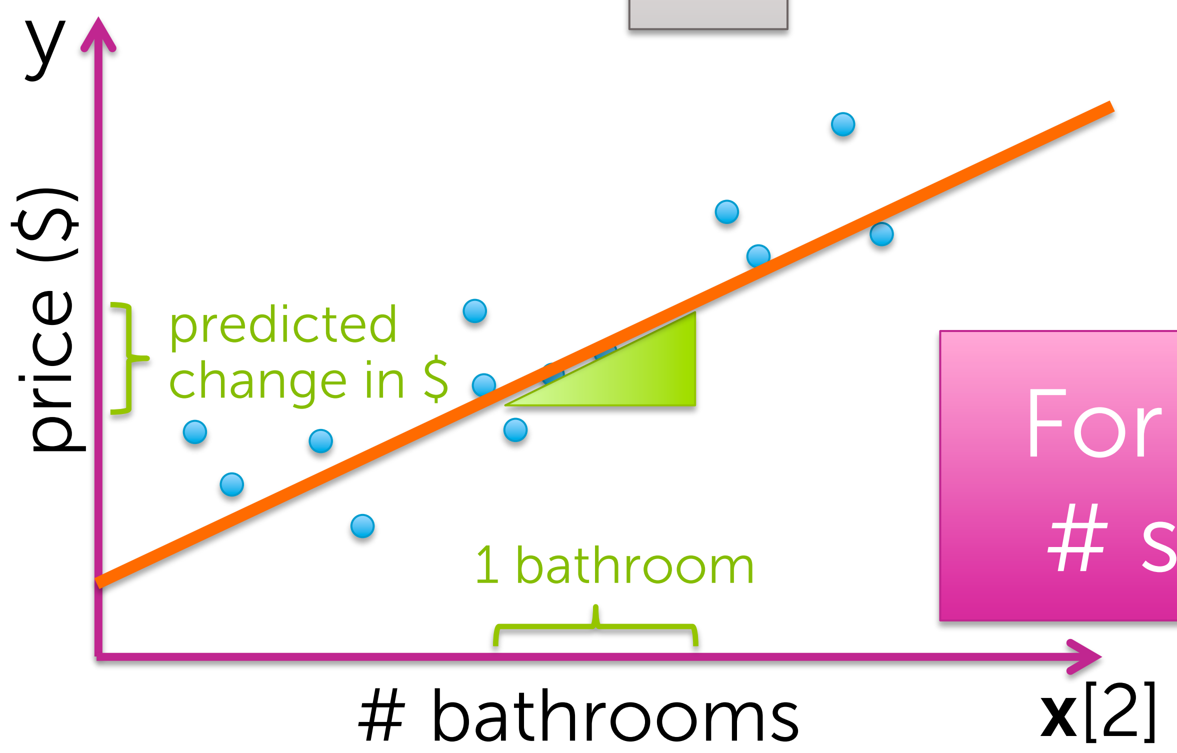
fix



Interpreting the coefficients – Two linear features

$$\hat{y} = \hat{w}_0 + \hat{w}_1 \boxed{\mathbf{x}[1]} + \hat{w}_2 \mathbf{x}[2]$$

fix

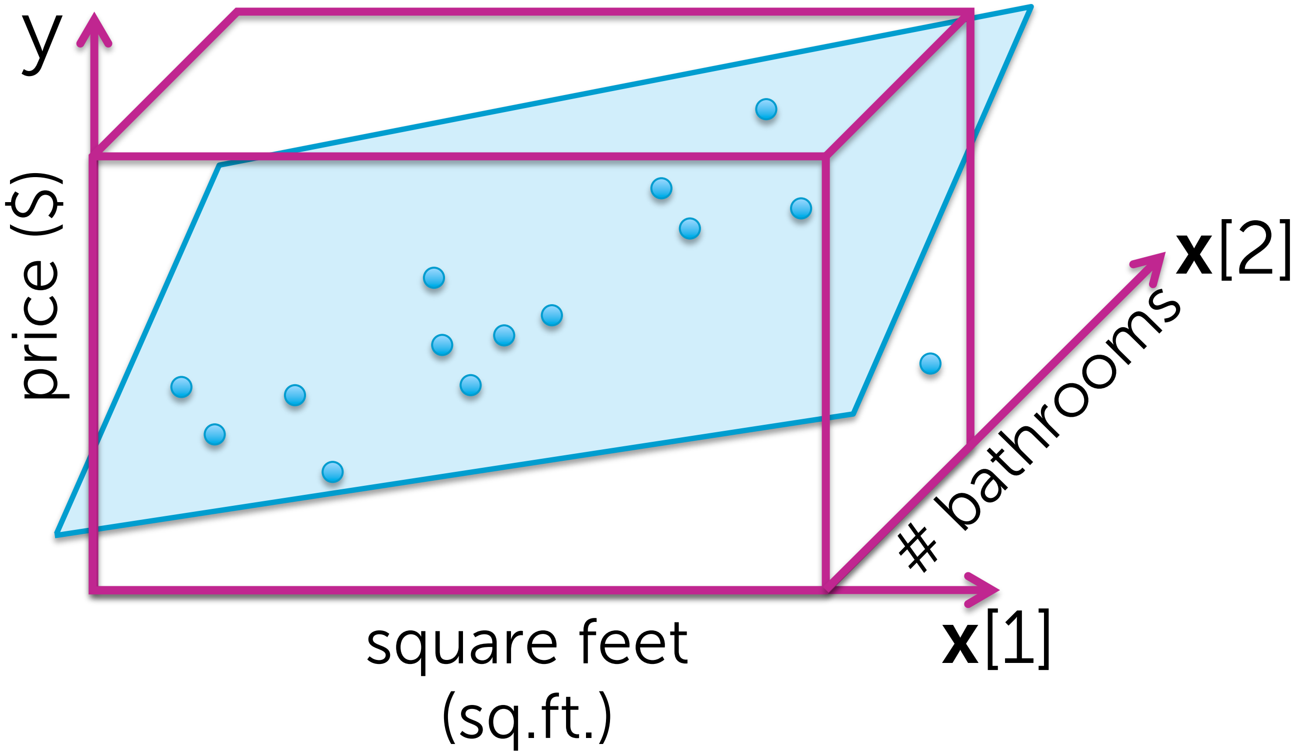


For fixed
sq.ft.!

Interpreting the coefficients – Multiple linear features

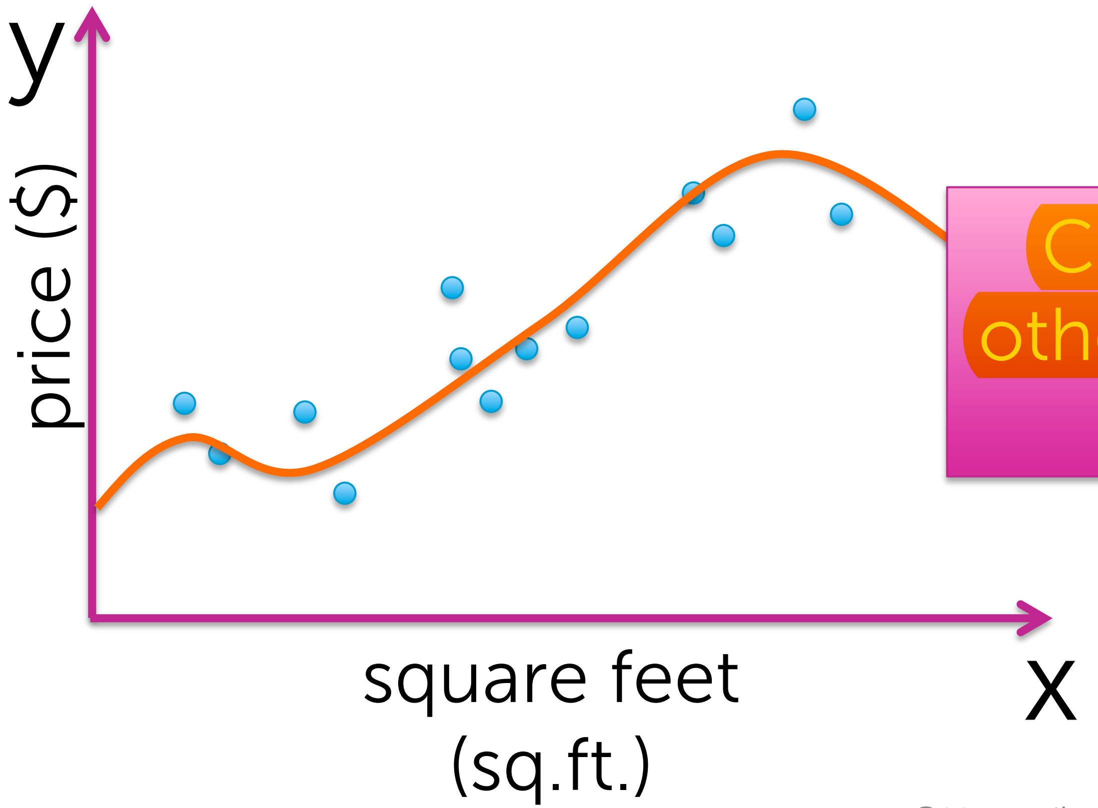
$$\hat{y} = \hat{w}_0 + \hat{w}_1 \mathbf{x}[1] + \ldots + \hat{w}_j \mathbf{x}[j] + \ldots + \hat{w}_d \mathbf{x}[d]$$

fix fix fix fix



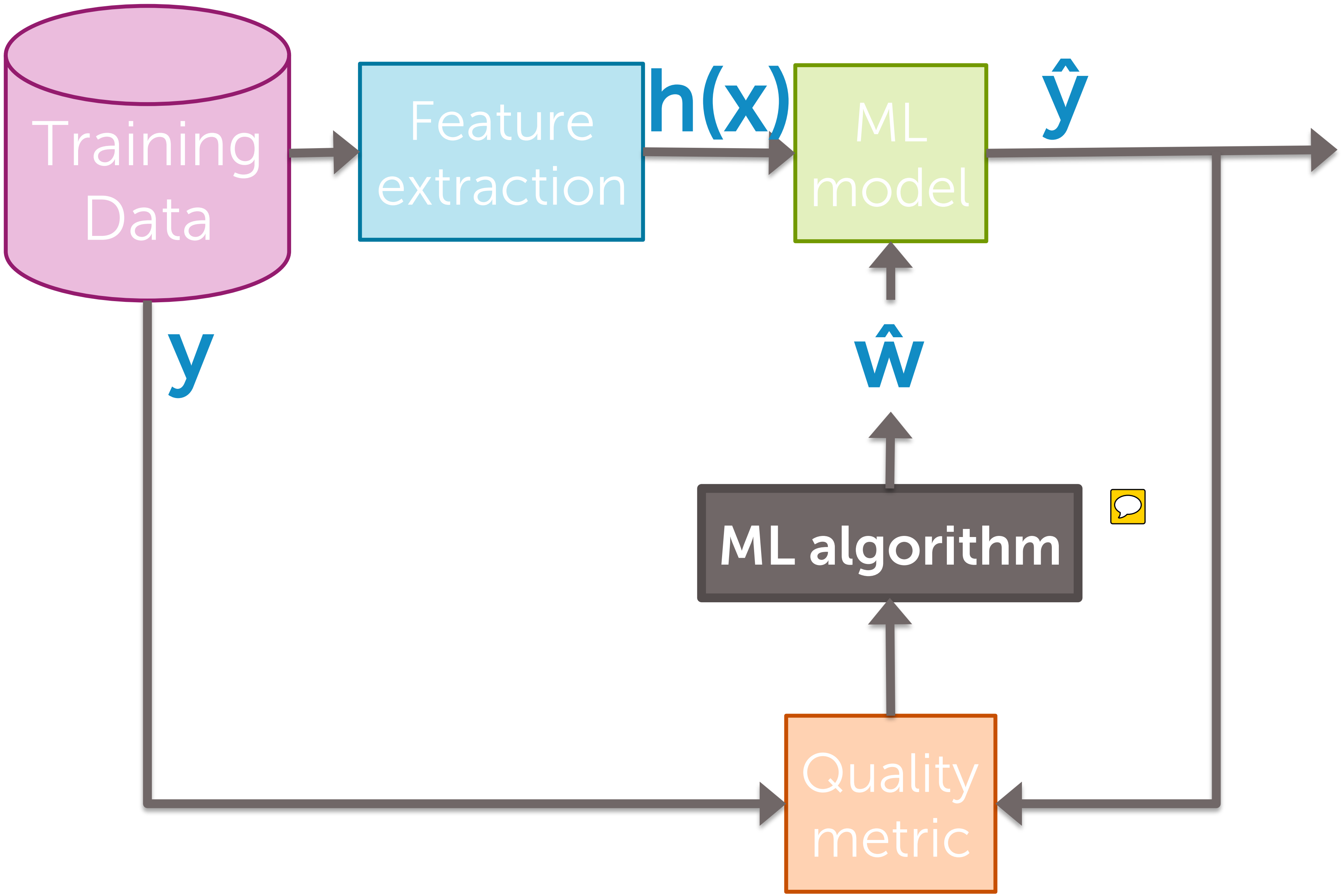
Interpreting the coefficients- Polynomial regression

$$\hat{y} = \hat{w}_0 + \hat{w}_1 x + \ldots + \hat{w}_j x^j + \ldots + \hat{w}_p x^p$$



Can't hold other features fixed!

Fitting D-dimensional curves


Training Data
Feature extraction
h(x)
ML model
ŷ
y
ŵ
ML algorithm
Quality metric

Step 1:
Rewrite the regression model

Rewrite in matrix notation

For observation i

$$y_i = \sum_{j=0}^{D} w_j \, h_j(\mathbf{x}_i) + \varepsilon_i$$

$$\boxed{y_i} = \overbrace{\begin{bmatrix} w_0 & w_1 & w_2 & \cdots & w_D \end{bmatrix}}^{w^T} \underbrace{\begin{bmatrix} h_0(x_i) \\ h_1(x_i) \\ h_2(x_i) \\ \vdots \\ h_D(x_i) \end{bmatrix}}_{h(x_i)} \; \square \; = \overbrace{\begin{bmatrix} h_0(x_i) & h_1(x_i) & \cdots & h_D(x_i) \end{bmatrix}}^{h^T(x_i)} + \varepsilon_i \; \overbrace{\begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_D \end{bmatrix}}^{w} \; \square$$

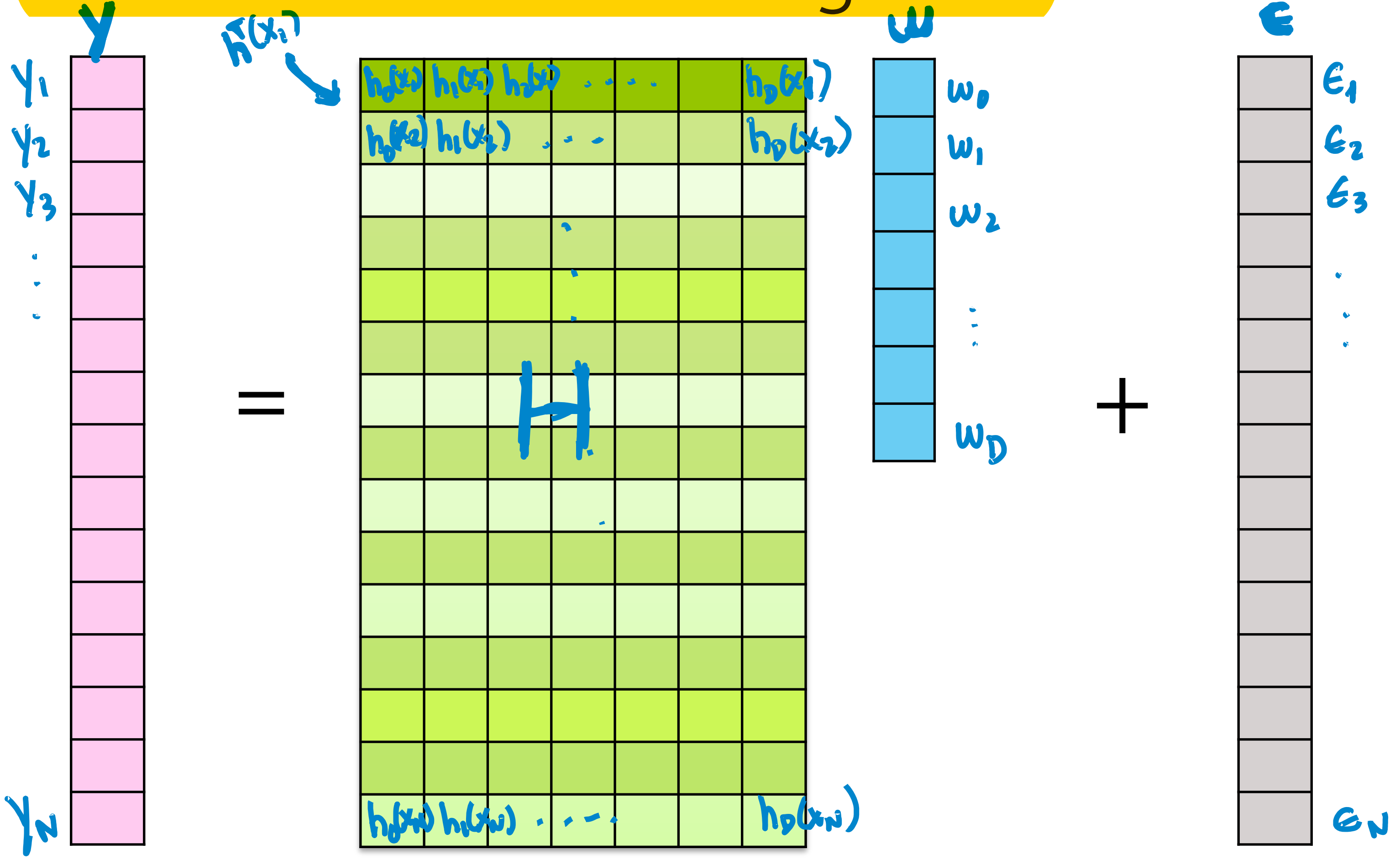
$$= \underbrace{w_0 h_0(x_i) + w_1 h_1(x_i) + \ldots + w_D h_D(x_i)}_{\text{scalar}} + \epsilon_i$$

$$= w^T h(x_i) + \epsilon_i$$

$$= h_0(x) w_0 + h_1(x_i) w_1 + \ldots + h_D(x_i) w_D + \epsilon_i$$

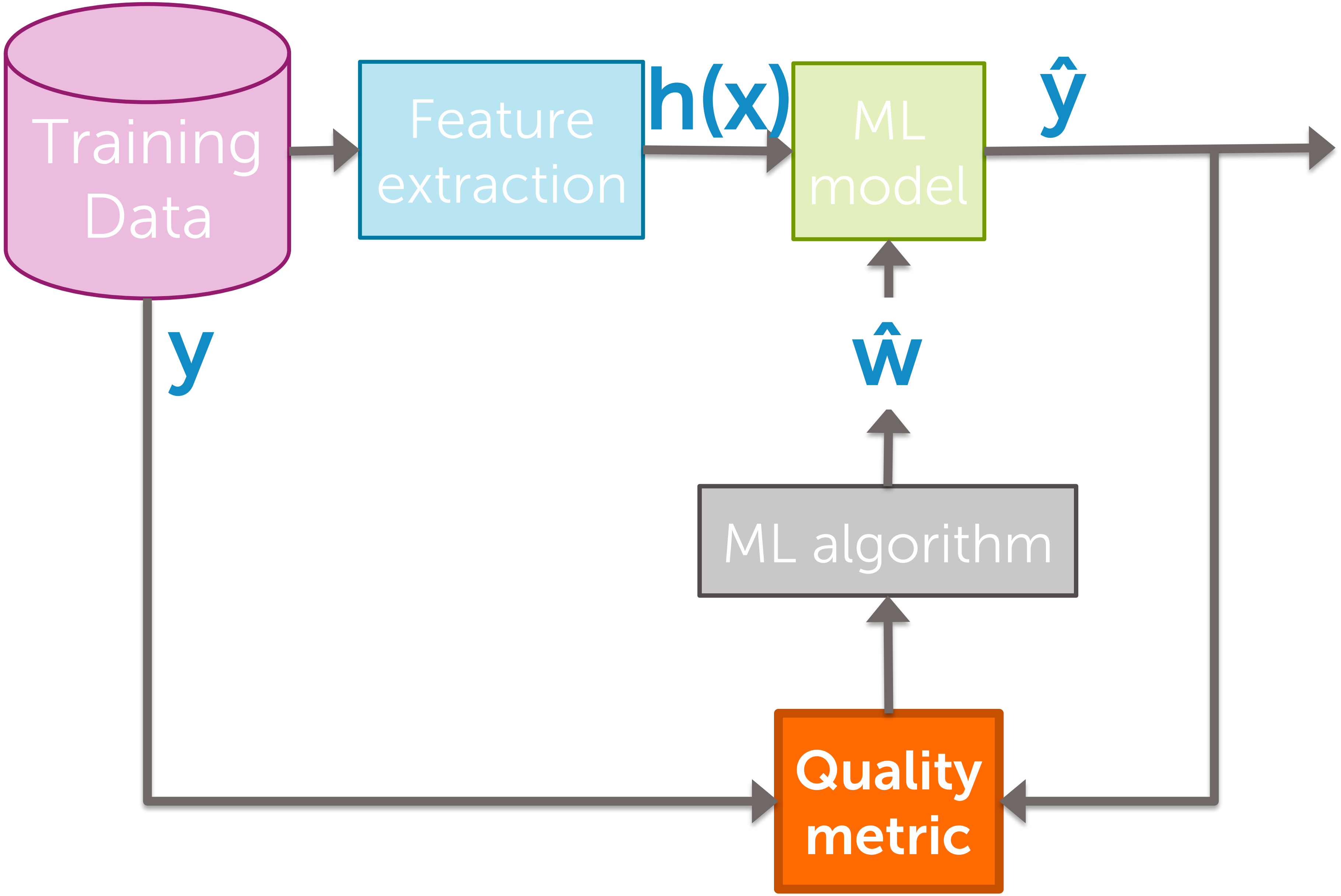
Rewrite in matrix notation

For all observations together



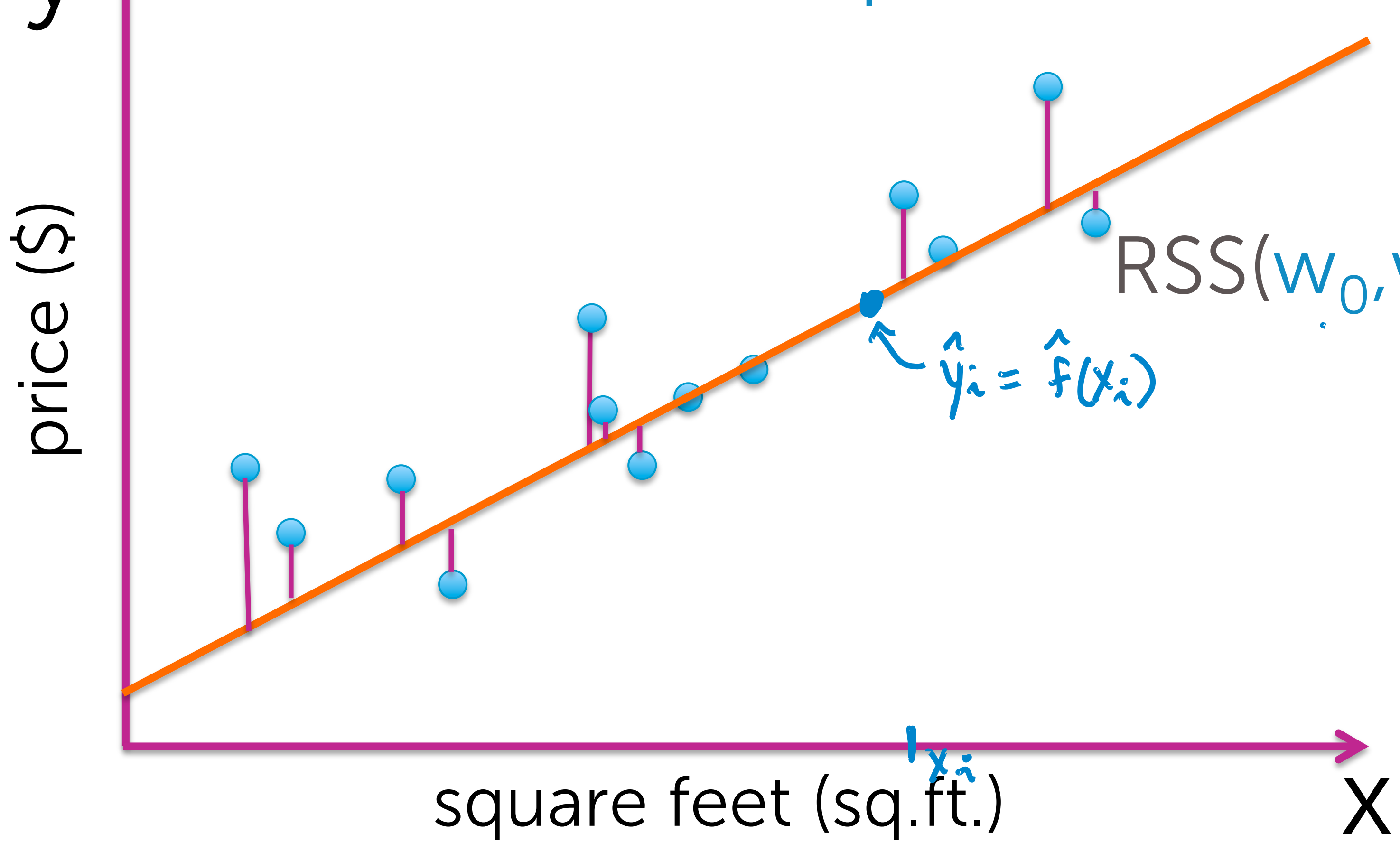
$$\Rightarrow \boxed{y = Hw + \epsilon}$$

Step 2:
Compute the cost


Training Data
Feature extraction
h(x)
ML model
ŷ
y
ŵ
ML algorithm
Quality metric

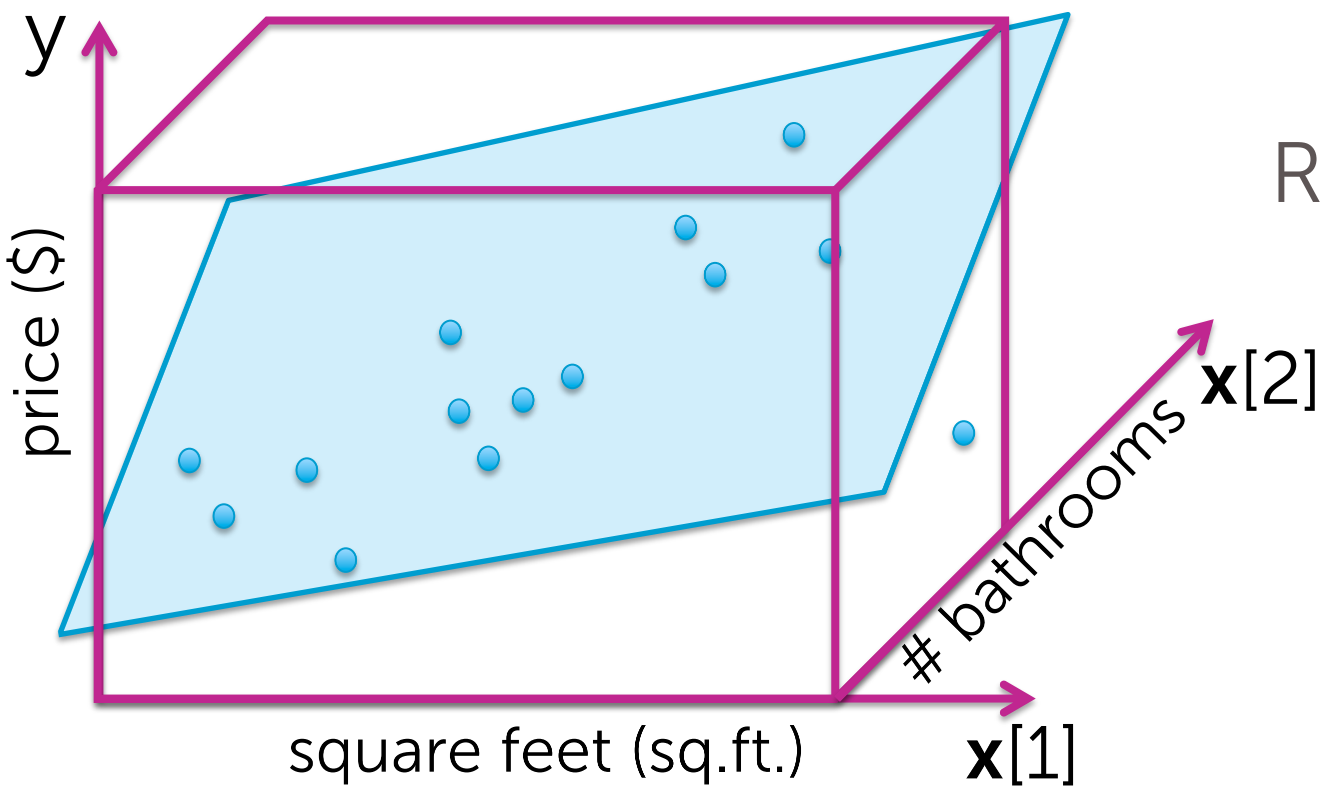
"Cost" of using a given line

Residual sum of squares (RSS)



$$\mathrm{RSS}(w_0,w_1) = \sum_{i=1}^{N} (y_i - [\underbrace{w_0 + w_1 x_i}_{\hat{y}_i(w_0,w_1)}])^2$$

RSS for multiple regression



$$\mathrm{RSS}(\underline{\mathbf{w}}) = \sum_{i=1}^{N} (y_i - \underbrace{h^T(x_i)\,w}_{\hat{y}_i(w)})^2$$

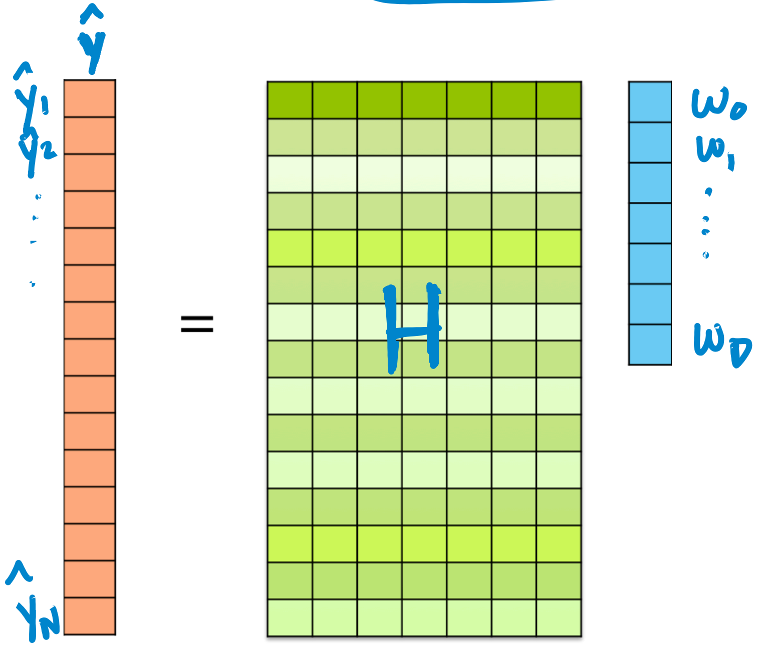
$$\hat{y}_i = \underbrace{\begin{bmatrix} h_0(x_i) & h_1(x_i) & \cdots & h_D(x_i) \end{bmatrix}}_{h^T(x_i)} \underbrace{\begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_D \end{bmatrix}}_{w}$$

RSS in matrix notation

$$\mathrm{RSS}(\mathbf{w}) = \sum_{i=1}^{N}(y_i - h(\mathbf{x}_i)^T\mathbf{w})^2$$

$$= (y-Hw)^T(y-Hw)$$

Why? (part 1)



$$\hat{y} = Hw$$

$$(y-Hw) = (y-\hat{y}) = \begin{bmatrix} \text{residual}_1 \\ \text{residual}_2 \\ \vdots \\ \text{residual}_N \end{bmatrix}$$

$\text{residual}_i = y_i - \hat{y}_i$

RSS in matrix notation

$$RSS(\mathbf{w}) = \sum_{i=1}^{N} (y_i - h(\mathbf{x}_i)^T \mathbf{w})^2$$

$$= (\mathbf{y}-\mathbf{H}\mathbf{w})^T(\mathbf{y}-\mathbf{H}\mathbf{w})$$

Why? (part 2)

$$\begin{bmatrix} \text{residual}_1 & \text{residual}_2 & \text{residual}_3 & \cdots & \text{residual}_N \end{bmatrix} \begin{bmatrix} \text{residual}_1 \\ \text{residual}_2 \\ \text{residual}_3 \\ \cdots \\ \text{residual}_N \end{bmatrix} = (\text{residual}_1^2 + \text{residual}_2^2 + \ldots + \text{residual}_N^2)$$

$$= \sum_{i=1}^{N} \text{residual}_i^2$$

$$\triangleq RSS(w)$$

Step 3:
Take the gradient

Gradient of RSS

$$\nabla \text{RSS}(\mathbf{w}) = \nabla[(\mathbf{y}-\mathbf{H}\mathbf{w})^T(\mathbf{y}-\mathbf{H}\mathbf{w})]$$

$$= -2\mathbf{H}^T(\mathbf{y}-\mathbf{H}\mathbf{w})$$

Why? By analogy to 1D case:

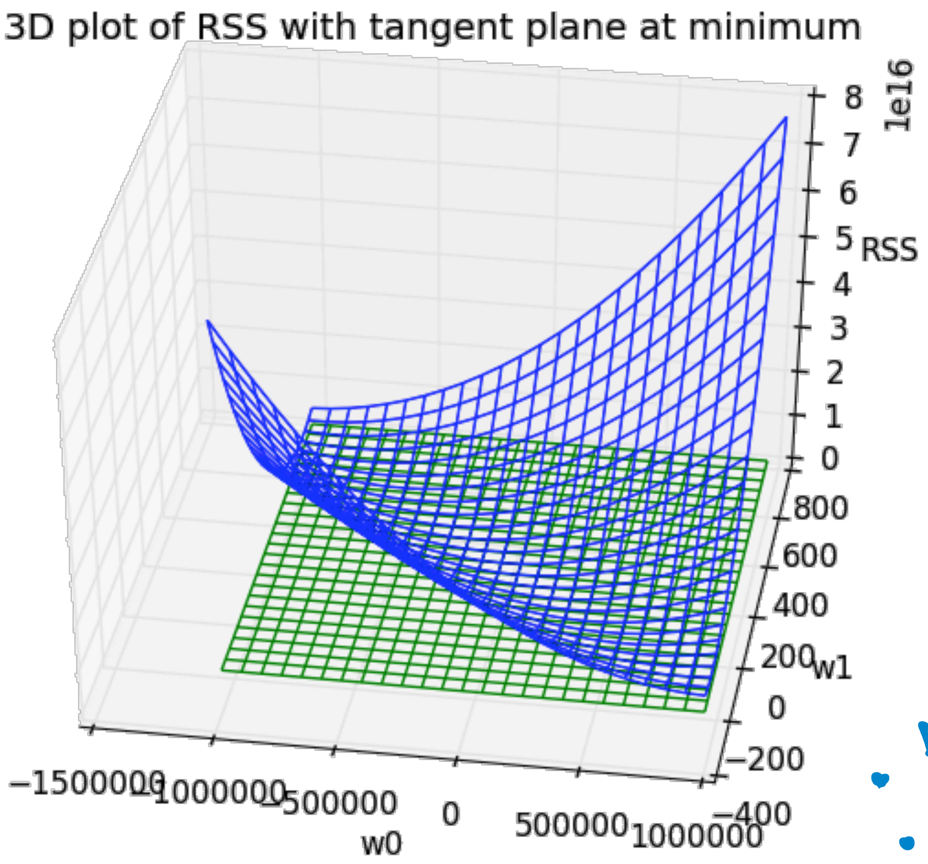
$$\frac{d}{dw}(y-hw)(y-hw) = \frac{d}{dw}(y-hw)^2 = 2\cdot(y-hw)^1(-h)$$

$$= -2h(y-hw)$$

scalars

Step 4, Approach 1:
Set the gradient = 0

Closed-form solution



∇RSS($\mathbf{w}$) = -2$\mathbf{H}^T$($\mathbf{y}$-$\mathbf{Hw}$) = 0

Solve for $\mathbf{w}$:

$$-\cancel{2}H^Ty + \cancel{2}H^TH\hat{w} = 0$$

$$H^TH\hat{w} = H^Ty$$

$$\underbrace{(H^TH)^{-1}H^TH}_{I}\hat{w} = (H^TH)^{-1}H^Ty$$

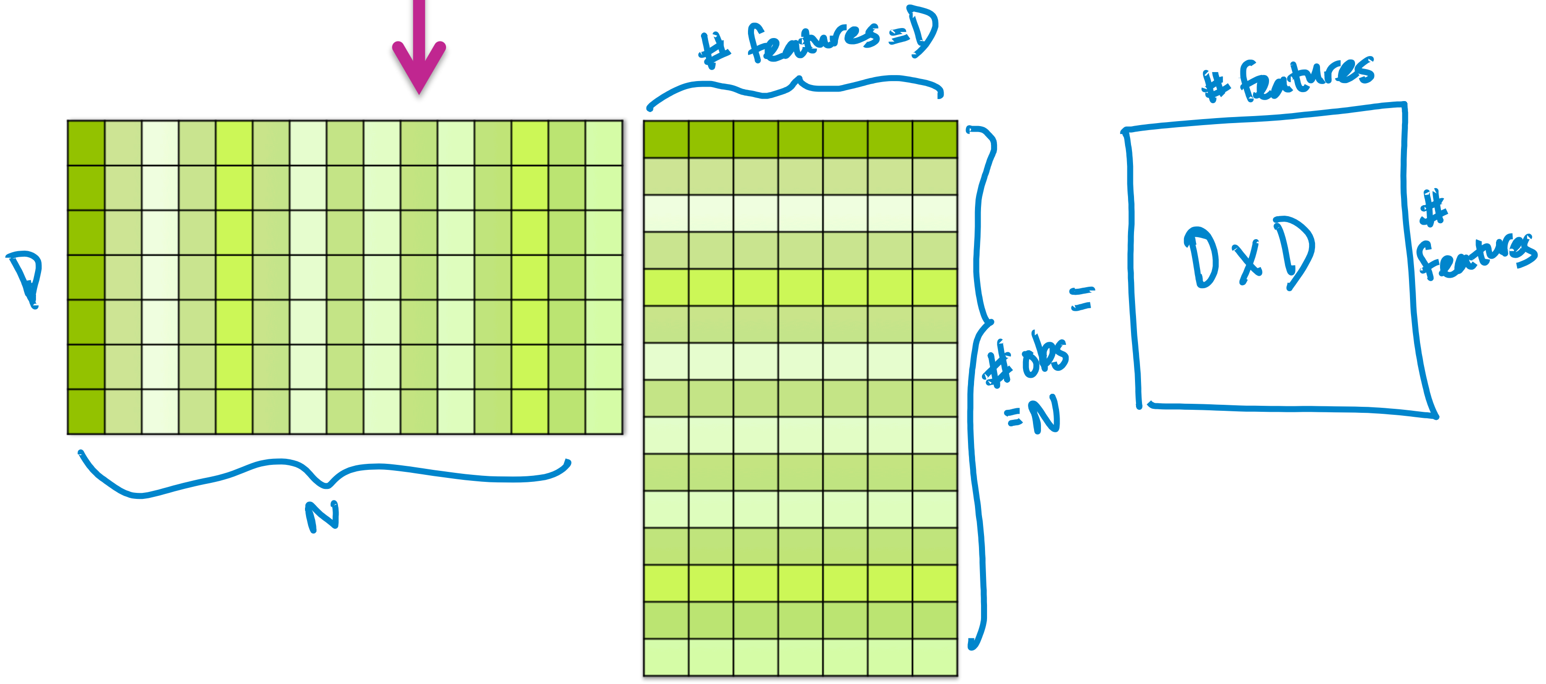
$$\hat{w} = (H^TH)^{-1}H^Ty$$

- $A^{-1}A = I$
- $Iv = v$

$IV = V$

Closed-form solution

$$\boxed{\hat{\mathbf{w}} = (\underbrace{\mathbf{H}^T\mathbf{H}})^{-1}\mathbf{H}^T\mathbf{y}}$$



Invertible if:

really, # of linearly ind. observations

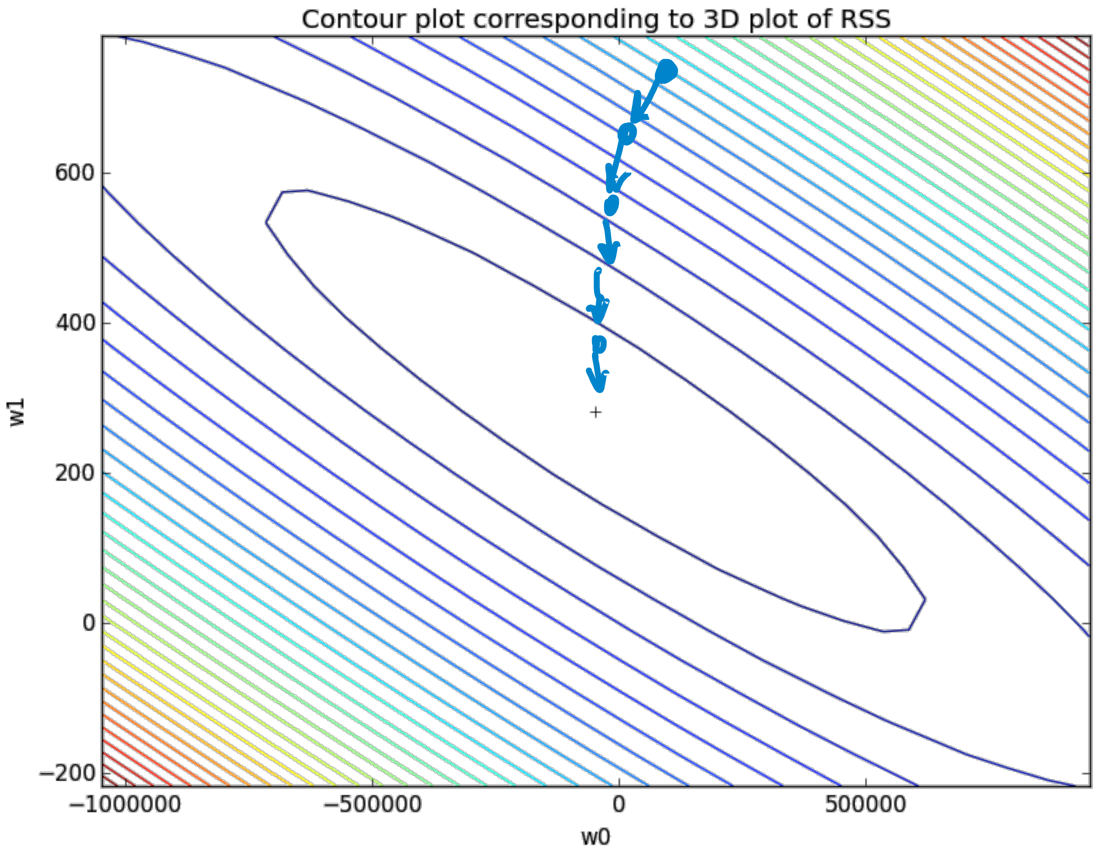
In most cases is N > D

Complexity of inverse:

$O(D^3)$

Step 4, Approach 2:
Gradient descent

Gradient descent



while not converged

$$\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} - \eta \underbrace{\nabla \text{RSS}(\mathbf{w}^{(t)})}_{-2\mathbf{H}^T(\mathbf{y}-\mathbf{H}\mathbf{w})}$$

$$\leftarrow w^{(t)} + 2\eta H^T(y - \underbrace{Hw^{(t)}}_{\hat{y}(w^{(t)})})$$

Feature-by-feature update

$$\mathrm{RSS}(\mathbf{w}) = \sum_{i=1}^{N}(y_i - h(\mathbf{x}_i)^T\mathbf{w})^2$$

$$= \sum_{i=1}^{N}\left(y_i - w_0h_0(x_i) - w_1h_1(x_i) - \ldots - w_Dh_D(x_i)\right)^2$$

Partial with respect to w_j

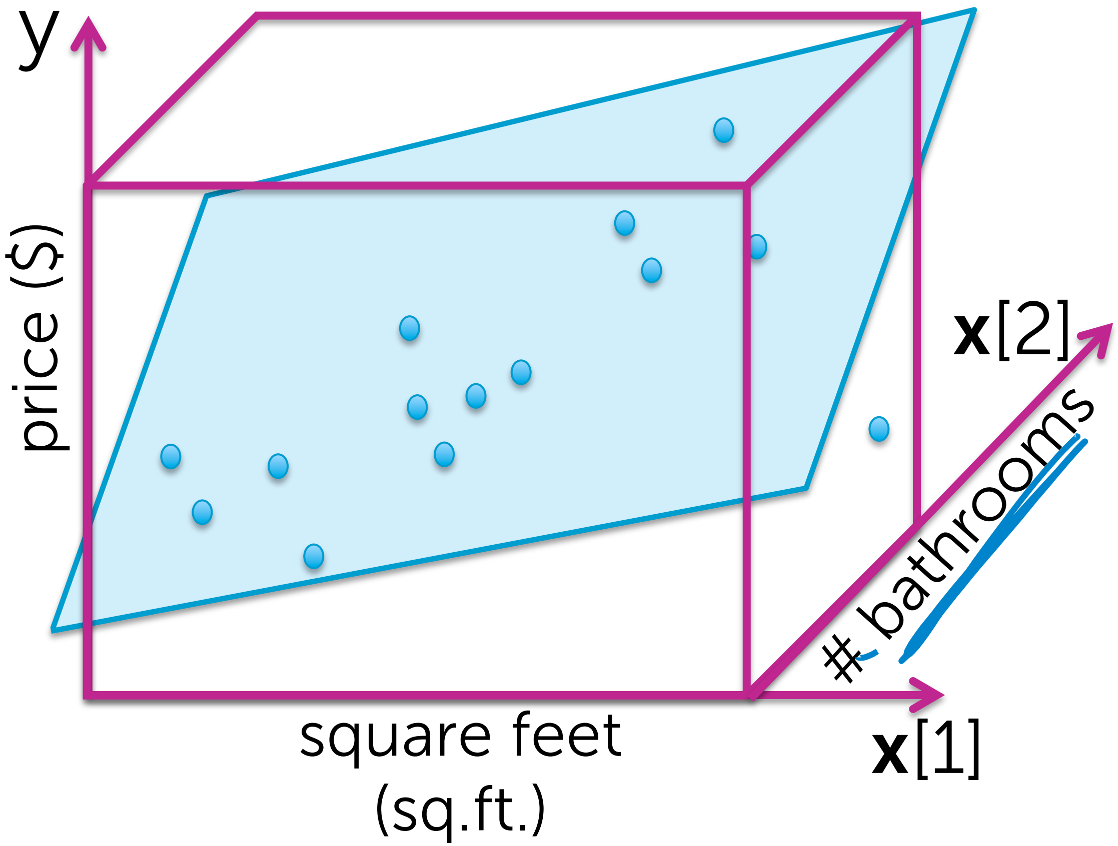
$$\sum_{i=1}^{N} 2\left(y_i - w_0h_0(x_i) - w_1h_1(x_i) \ldots - w_Dh_D(x_i)\right)^1 \cdot \left(\underline{-h_j(x_i)}\right)$$

$$= -2\sum_{i=1}^{N} h_j(x_i)\left(y_i - h(x_i)^T w\right)$$

Update to j^{th} feature weight:

$$w_j^{(t+1)} \leftarrow w_j^{(t)} - \eta\left(-2\sum_{i=1}^{N} h_j(x_i)\left(y_i - \underbrace{h^T(x_i)w^{(t)}}_{\hat{y}_i(w^{(t)})}\right)\right)$$

Interpreting elementwise



Update to j^{th} feature weight:

\# bathrooms

\# bath for house i

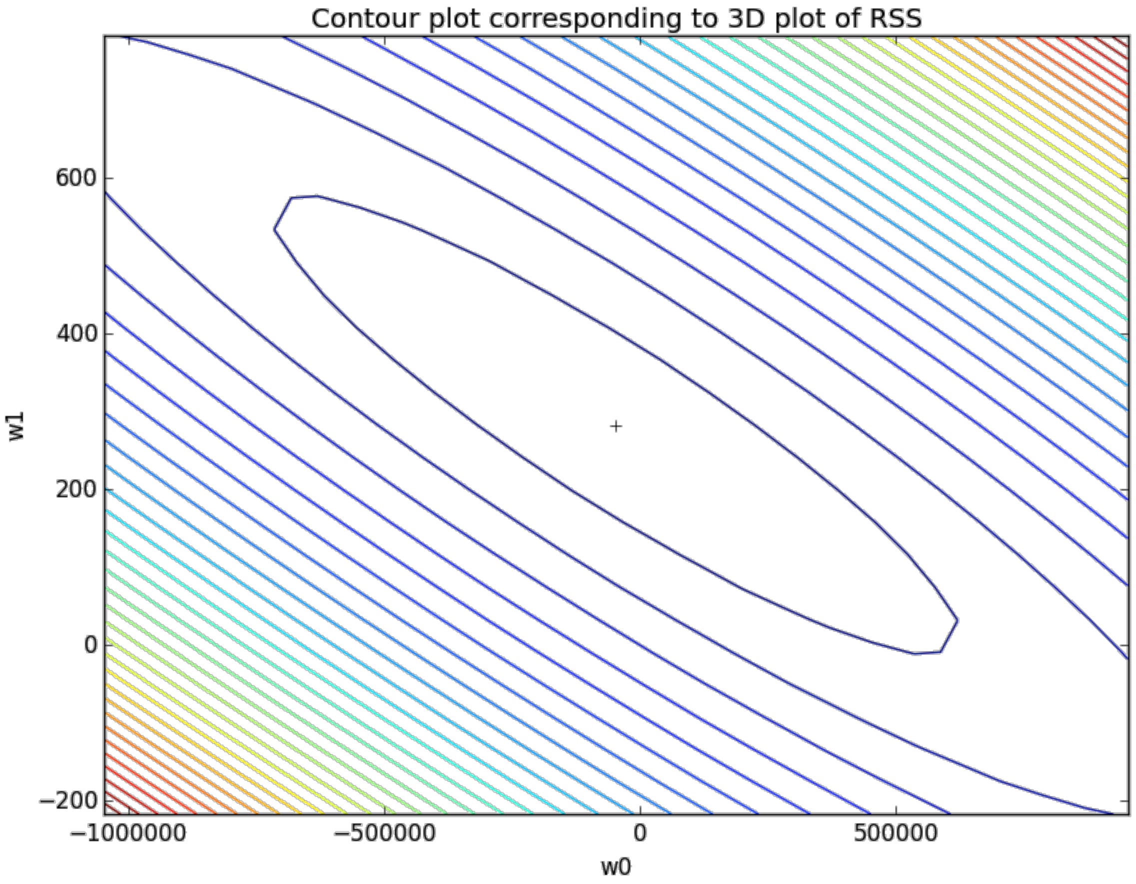
$$w_j^{(t+1)} \leftarrow w_j^{(t)} + 2\eta \sum_{i=1}^{N} h_j(\mathbf{x}_i)(y_i - \hat{y}_i(\mathbf{w}^{(t)}))$$

If underestimating impact of # bath ($\hat{w}_j^{(t)}$ is too small)

then $(y_i - \hat{y}_i(w^{(t)}))$ on average weighted by # bath will be positive

$\Rightarrow w_j^{(t+1)} > w_j^{(t)}$ (increase)

Summary of gradient descent for multiple regression



init $\mathbf{w}^{(1)}=0$ (or randomly, or smartly), $t=1$

tolerance

while $||\nabla \text{RSS}(\mathbf{w}^{(t)})|| > \varepsilon$ $\sqrt{\text{partial}[0]^2 + \ldots + \text{partial}[D]^2}$

for $j=0,\ldots,D$

$\text{partial}[j] = -2\sum_{i=1}^{N} h_j(\mathbf{x}_i)(y_i-\hat{y}_i(\mathbf{w}^{(t)}))$

$\mathbf{w}_j^{(t+1)} \leftarrow \mathbf{w}_j^{(t)} - \eta\ \text{partial}[j]$

$t \leftarrow t + 1$

An extremely useful algorithm

Summary for multiple linear regression

What you can do now…

- Describe polynomial regression
- Detrend a time series using trend and seasonal components
- Write a regression model using multiple inputs or features thereof
- Cast both polynomial regression and regression with multiple inputs as regression with multiple features
- Calculate a goodness-of-fit metric (e.g., RSS)
- Estimate model parameters of a general multiple regression model to minimize RSS:
  - In closed form
  - Using an iterative gradient descent algorithm
- Interpret the coefficients of a non-featurized multiple regression fit
- Exploit the estimated model to form predictions
- Explain applications of multiple regression beyond house price modeling