Simple Regression:

Linear regression with one input

Emily Fox & Carlos Guestrin

Machine Learning Specialization

University of Washington

Recall Task:
Predicting house prices

How much is my house worth?

How much is my house worth?

Look at recent sales in my neighborhood

- How much did they sell for?

Regression fundamentals: data, model, task

Data

input *output*

$(x_1 = \text{sq.ft.}, y_1 = \$)$

$(x_2 = \text{sq.ft.}, y_2 = \$)$

$(x_3 = \text{sq.ft.}, y_3 = \$)$

$(x_4 = \text{sq.ft.}, y_4 = \$)$

$(x_5 = \text{sq.ft.}, y_5 = \$)$

⋮

Data

input *output*

$(x_1 = \text{sq.ft.}, y_1 = \$)$

$(x_2 = \text{sq.ft.}, y_2 = \$)$

$(x_3 = \text{sq.ft.}, y_3 = \$)$

$(x_4 = \text{sq.ft.}, y_4 = \$)$

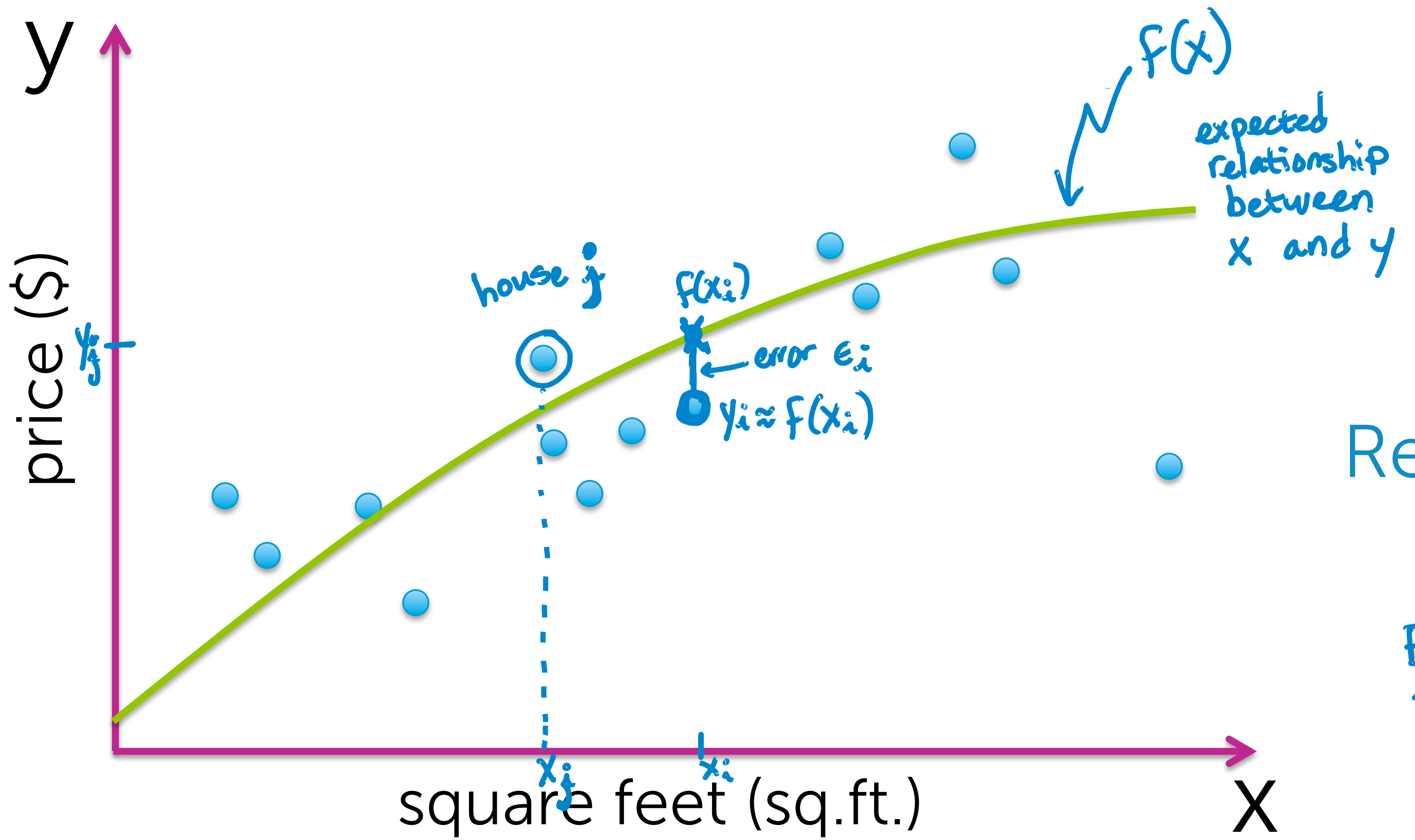
$(x_5 = \text{sq.ft.}, y_5 = \$)$

$\vdots$

Input vs. Output:

- y is the quantity of interest
- assume y can be predicted from x

Model –
How we *assume* the world works



Regression model:

$y_i = f(x_i) + \epsilon_i$

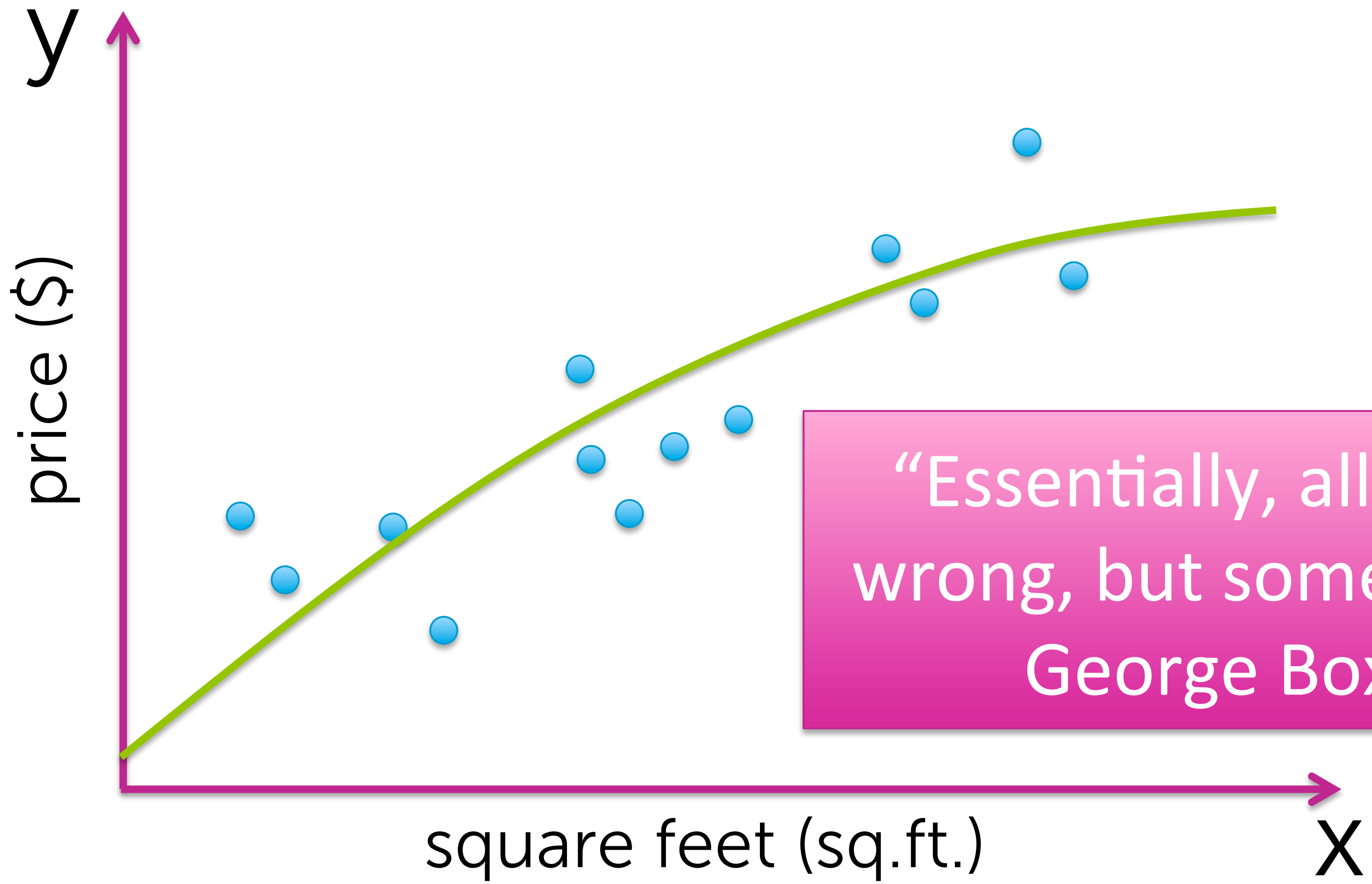
$E[\epsilon_i] = 0$ ← equally likely that error is + or −

↑ expected value

⇓

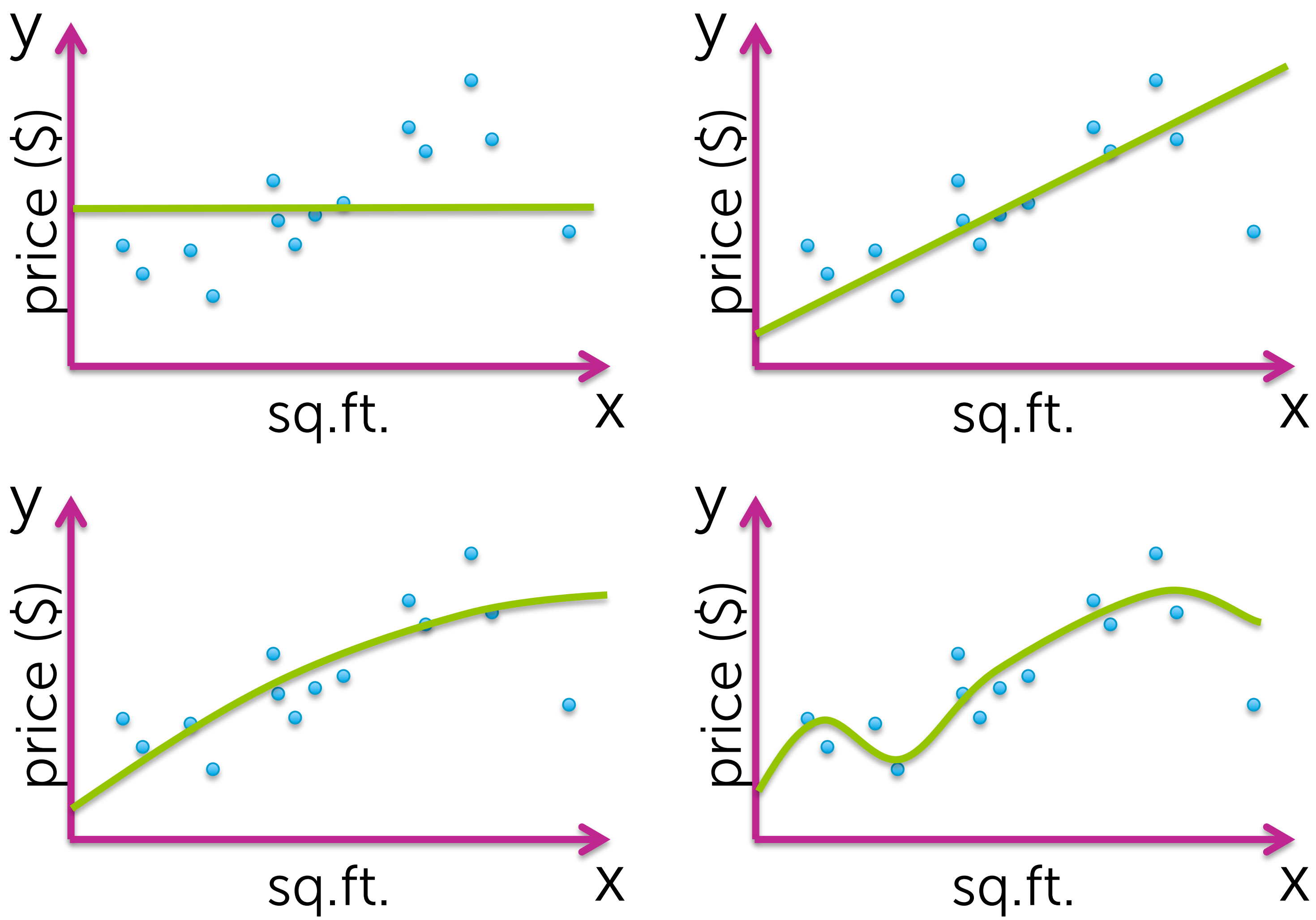
y_i is equally likely to be above or below $f(x_i)$

Model –
How we *assume* the world works

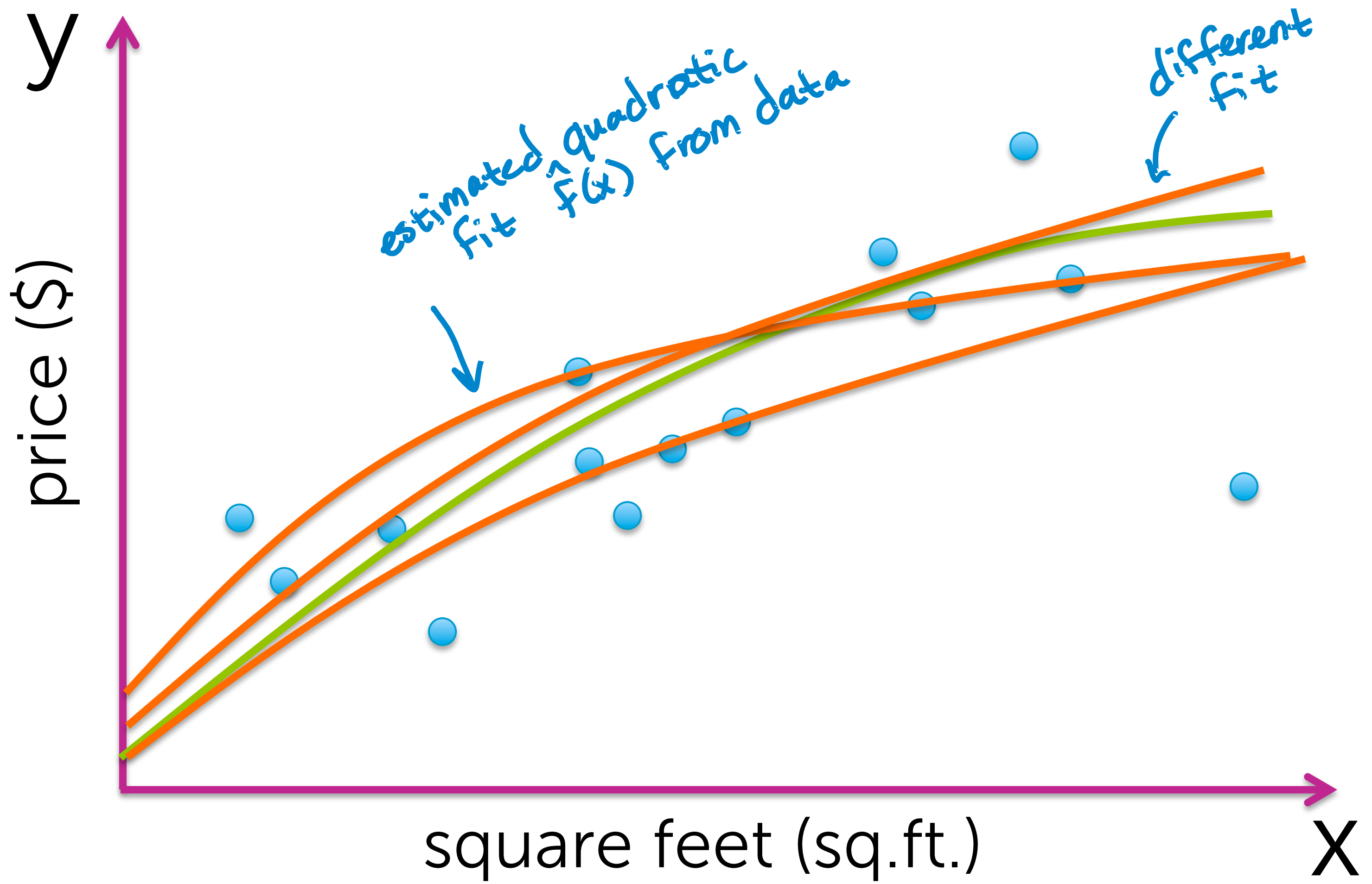


"Essentially, all models are wrong, but some are useful."
George Box, 1987.

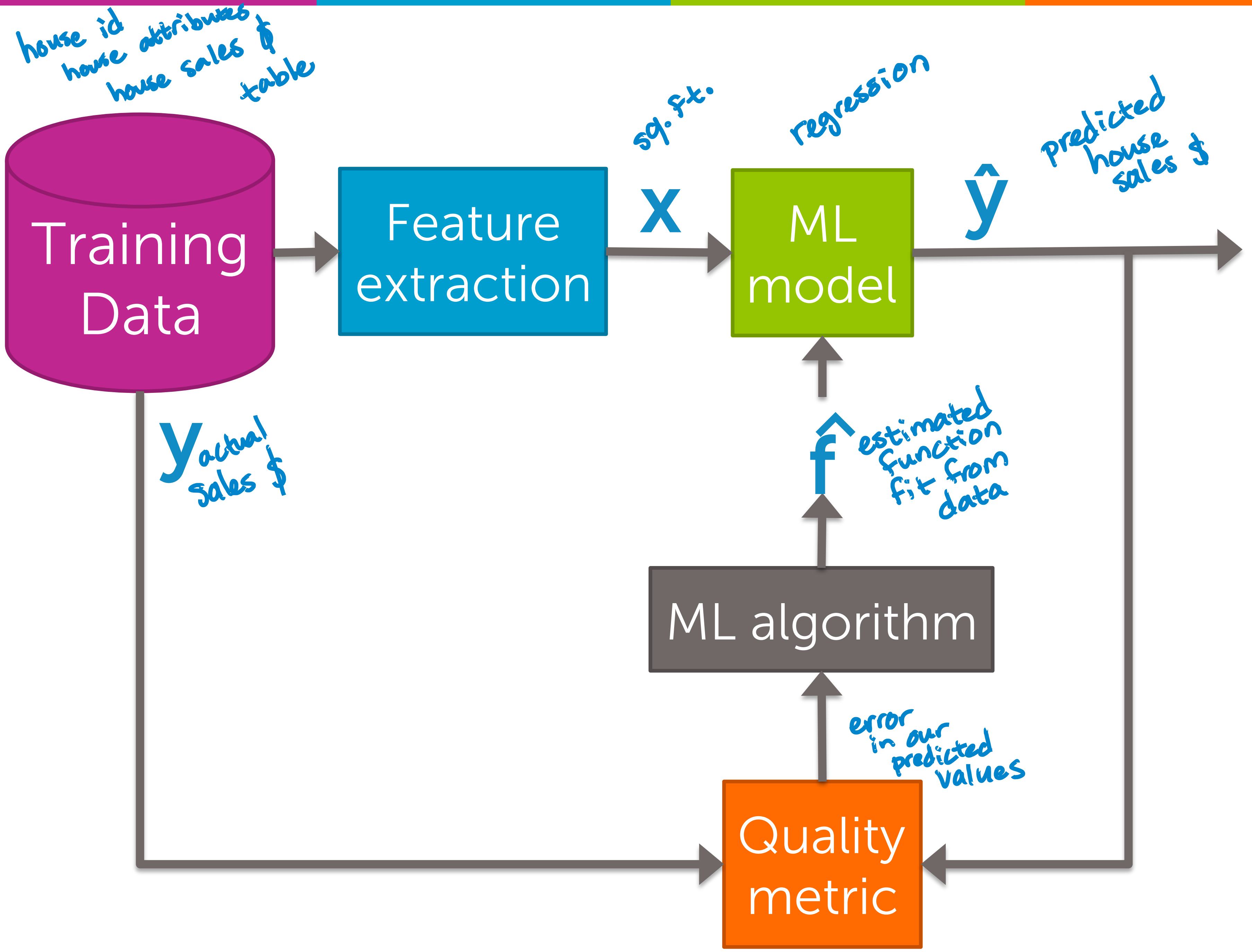
Task 1–
Which model f(x)?



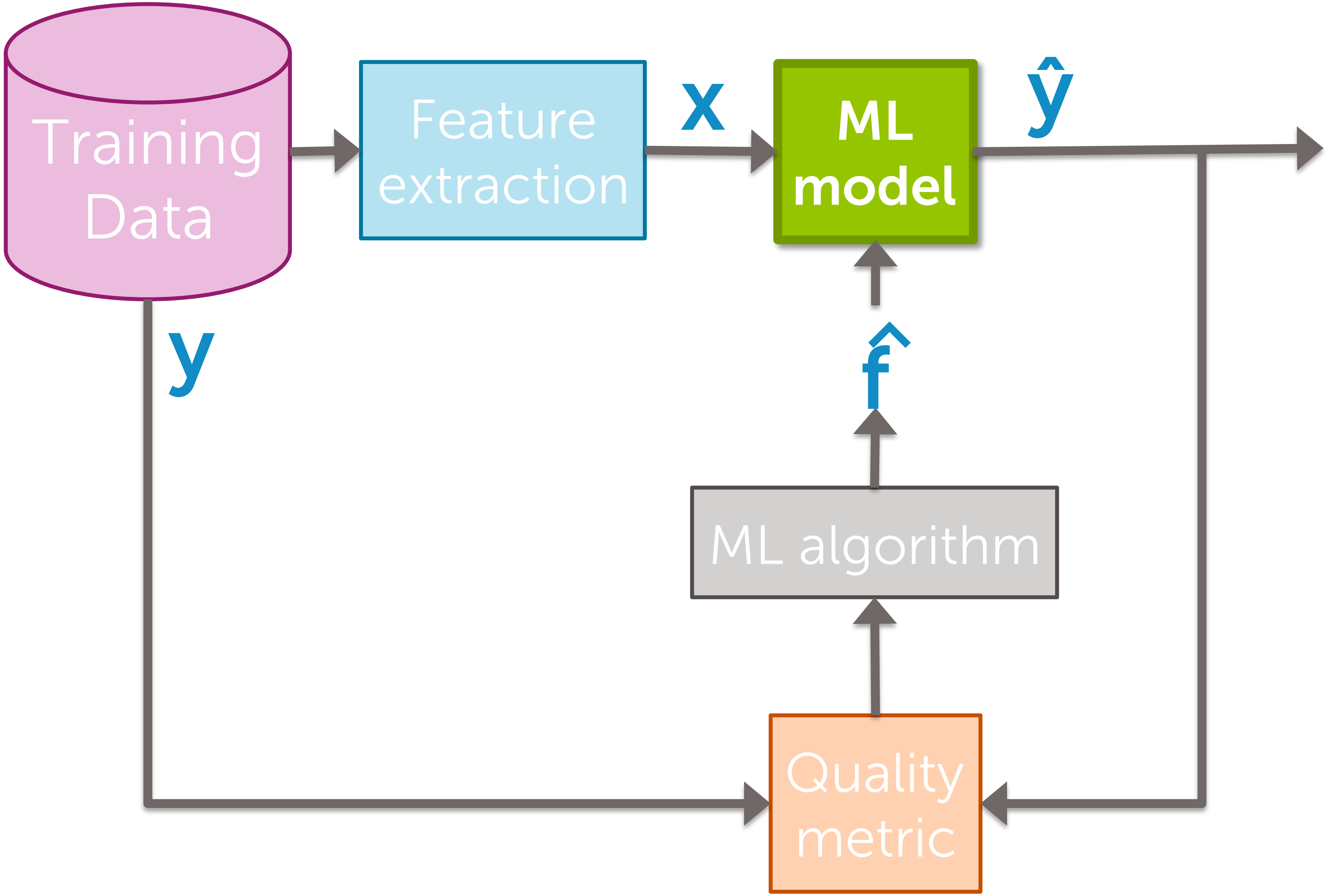
Task 2 – For a given model f(x), estimate function $\hat{f}(x)$ from data



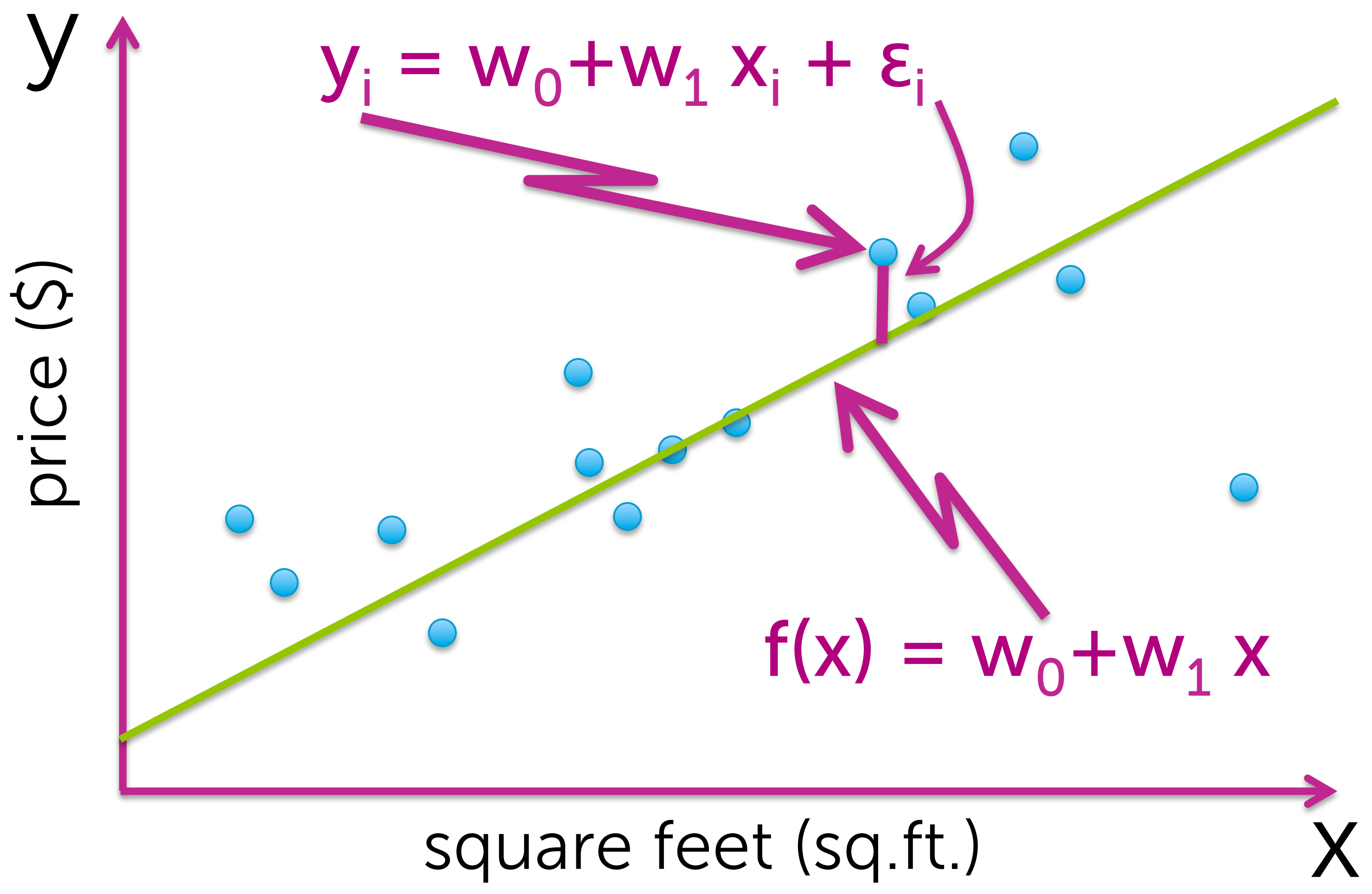
Assume model f(x) is a quadratic function


house id
house attributes
house sales $
table
Training Data
Feature extraction
sq. ft.
x
regression
ML model
ŷ
predicted house sales $
y actual sales $
f̂
estimated function fit from data
ML algorithm
error in our predicted values
Quality metric

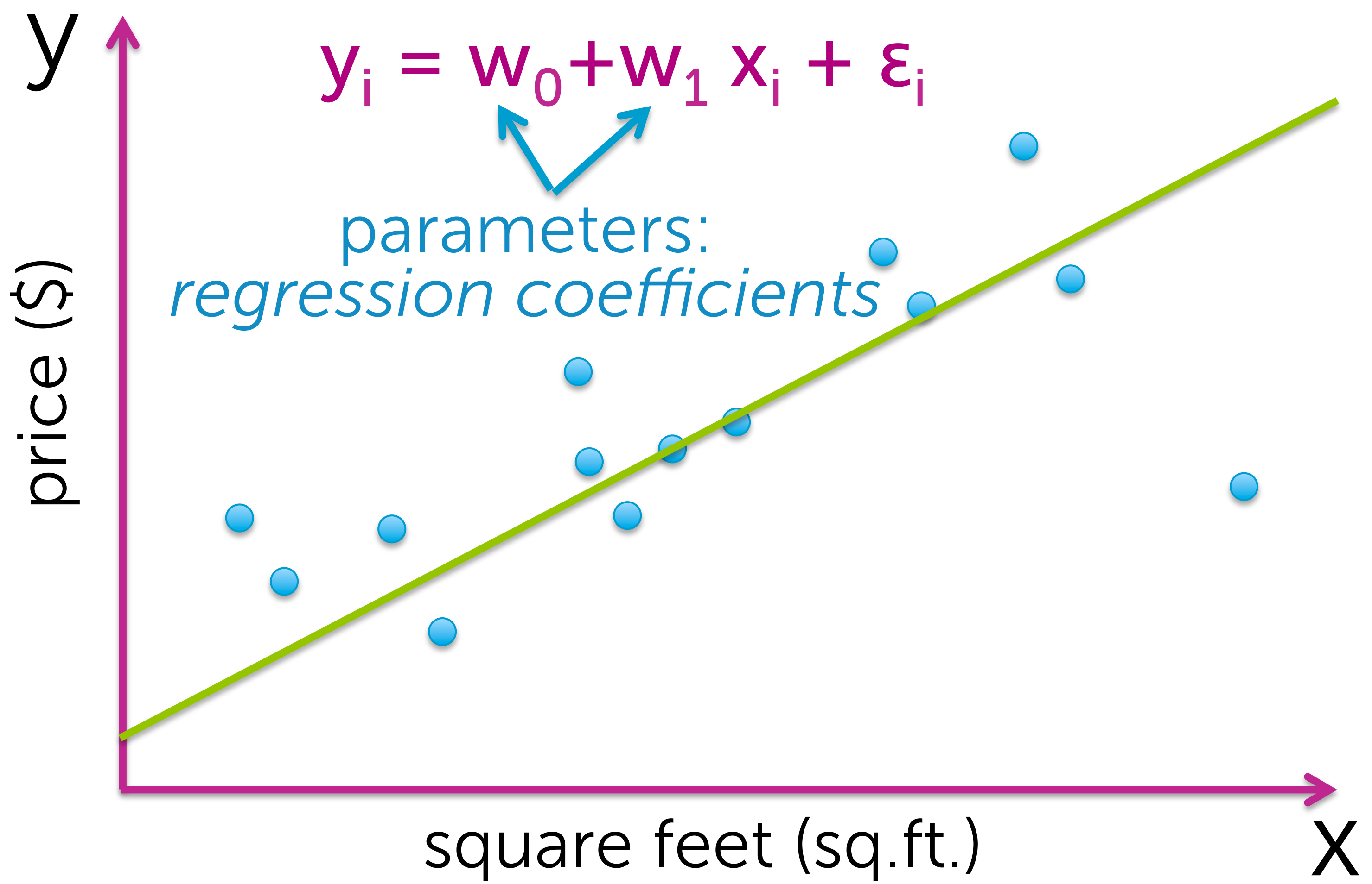
Simple linear regression


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

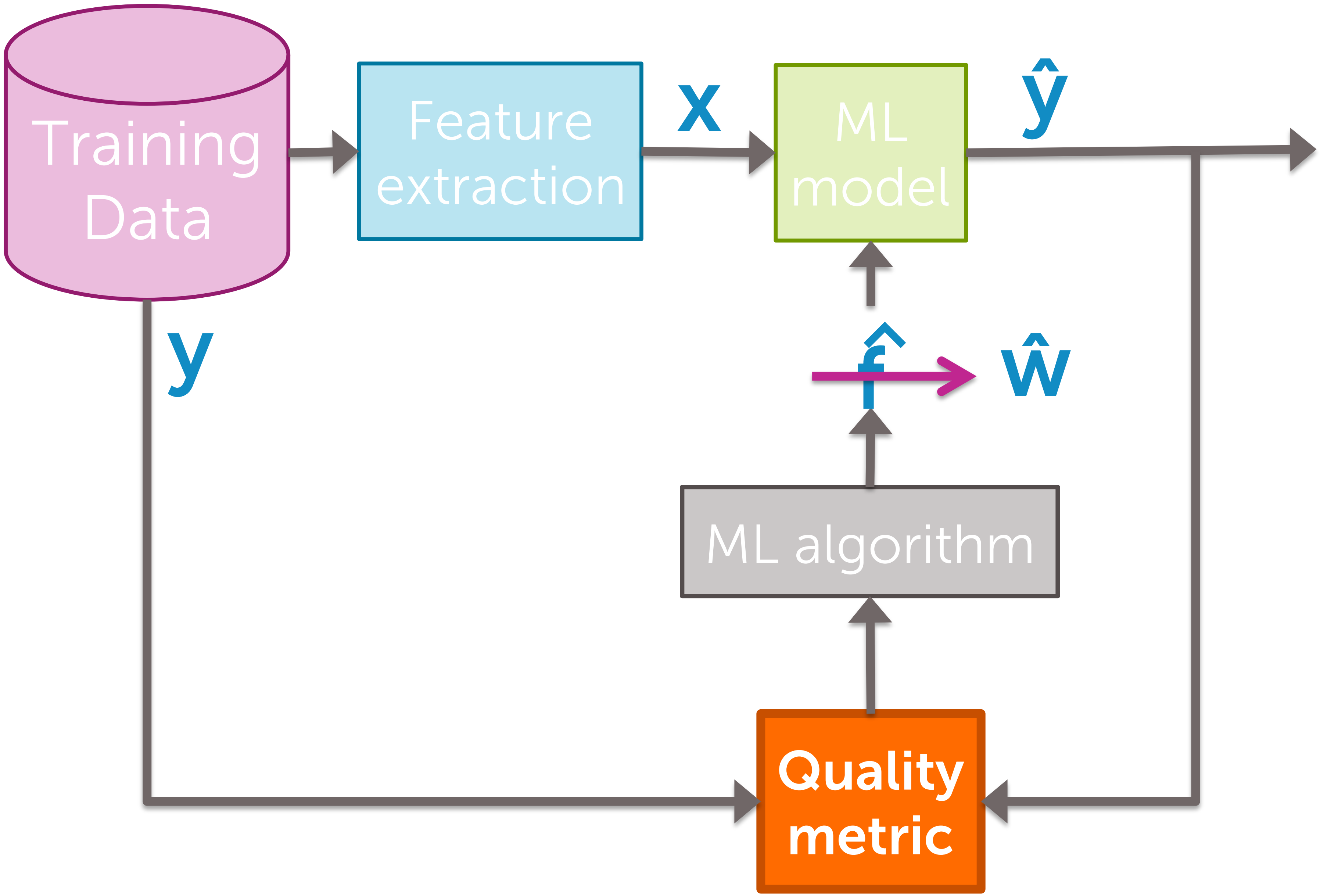
Simple linear regression model



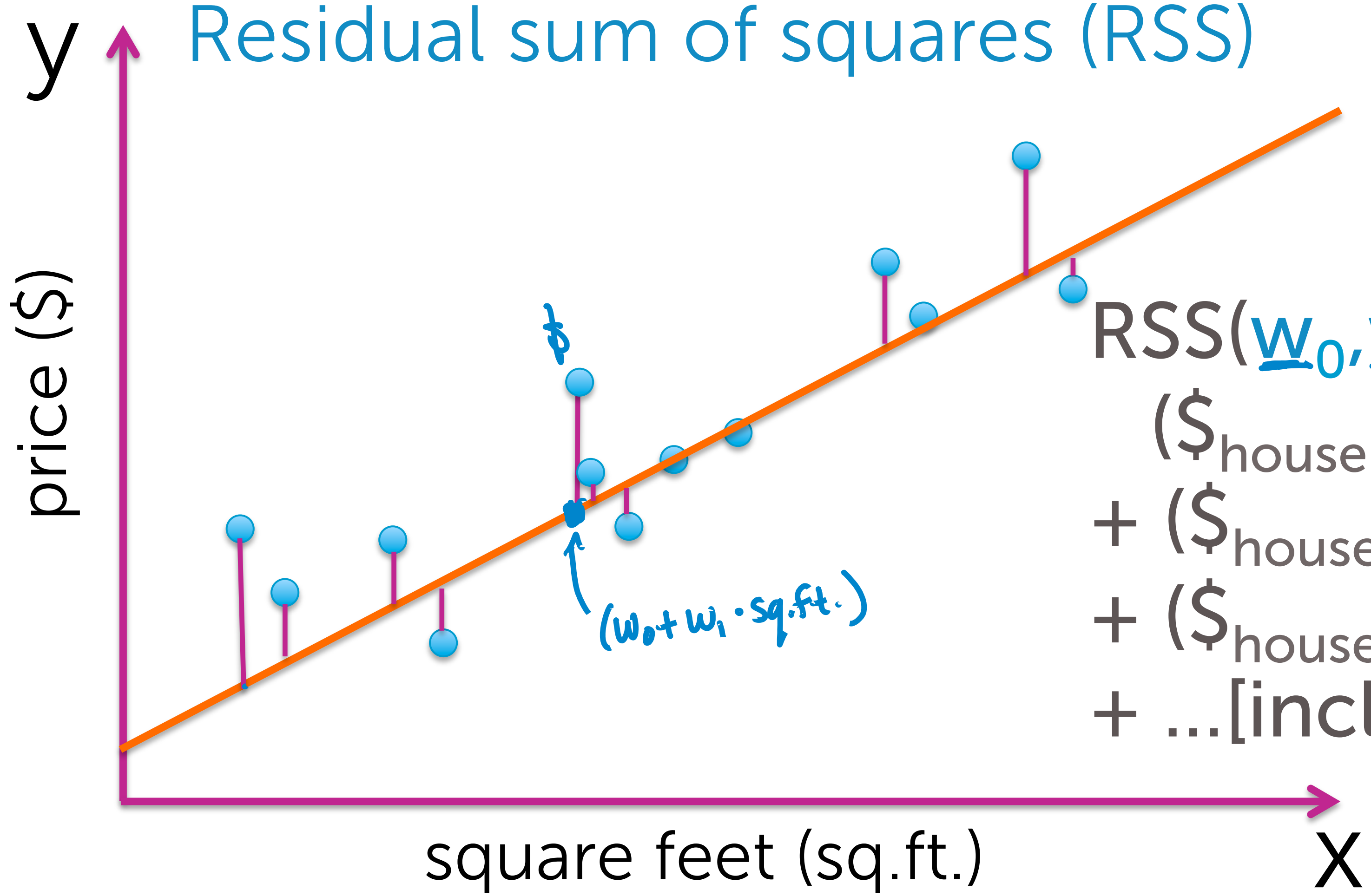
Simple linear regression model



Fitting a line to data


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

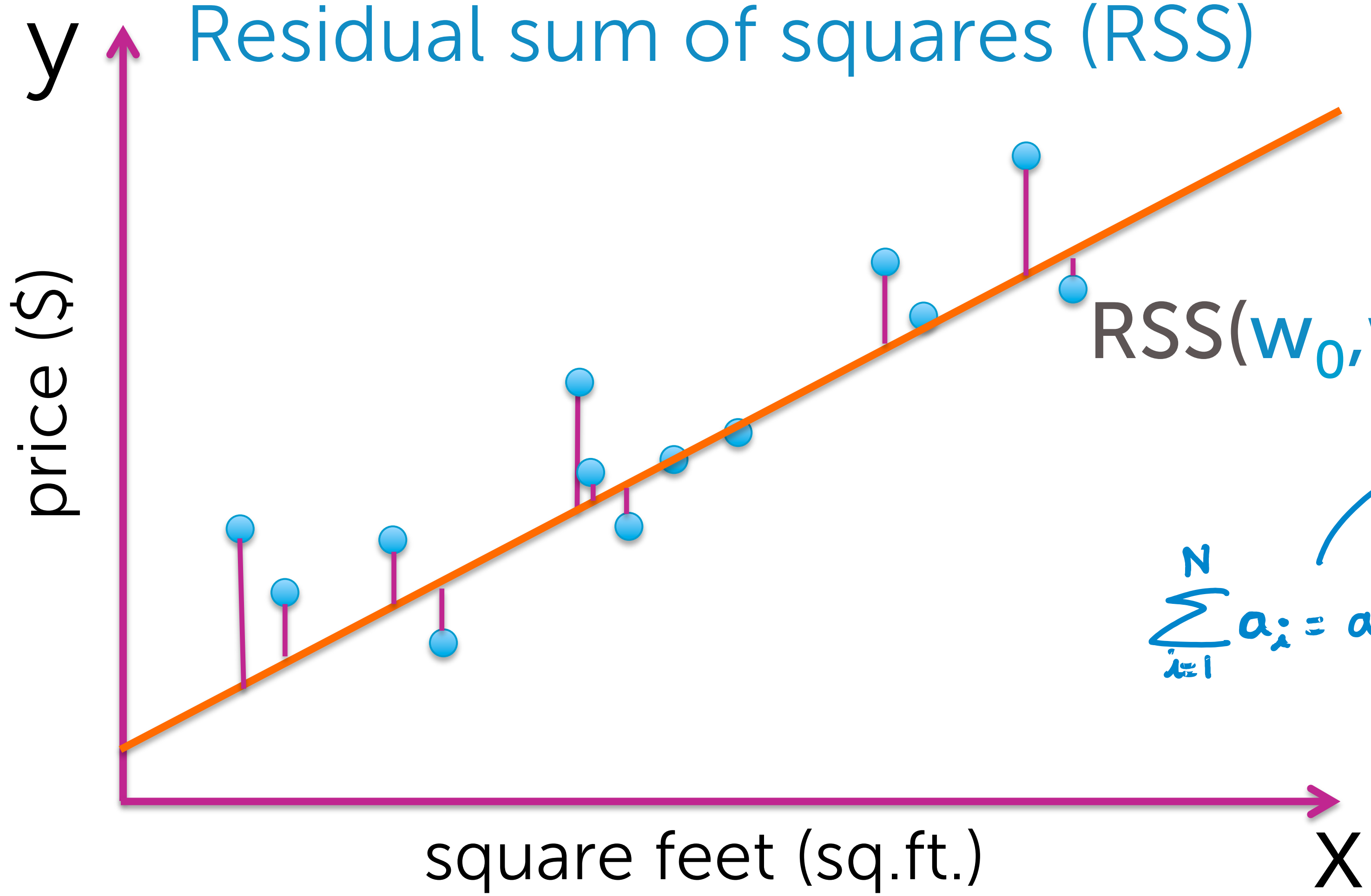
"Cost" of using a given line



$$RSS(w_0,w_1) =$$
$$(\$_{house\ 1}-[w_0+w_1 sq.ft._{house\ 1}])^2$$
$$+ (\$_{house\ 2}-[w_0+w_1 sq.ft._{house\ 2}])^2$$
$$+ (\$_{house\ 3}-[w_0+w_1 sq.ft._{house\ 3}])^2$$
+ ...[include all training houses]

"Cost" of using a given line

Residual sum of squares (RSS)



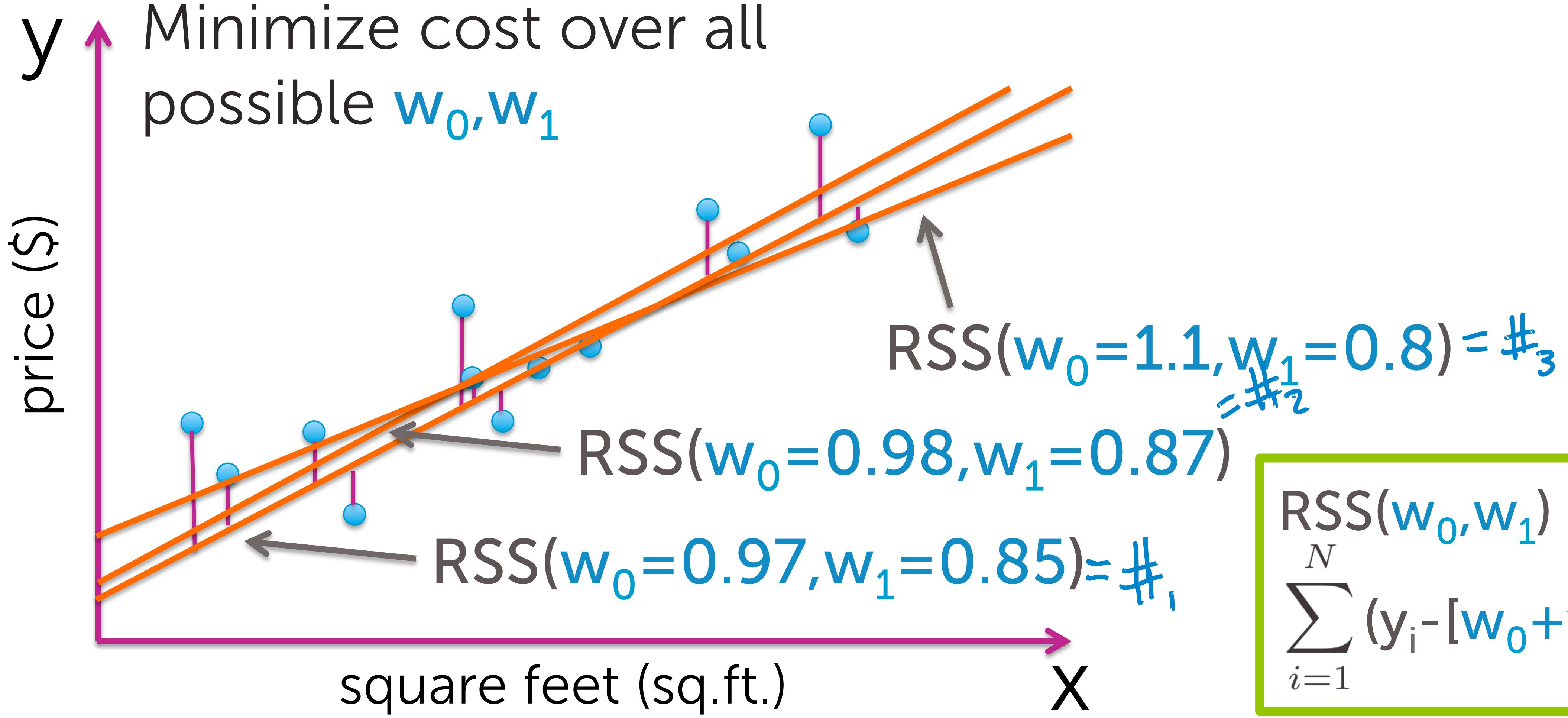
$$RSS(w_0,w_1) = \sum_{i=1}^{N} (y_i-[w_0+w_1x_i])^2$$

$$\sum_{i=1}^{N} a_i = a_1 + a_2 + \ldots + a_N$$

Here,

$$a_i = (y_i - [w_0+w_1x_i])^2$$

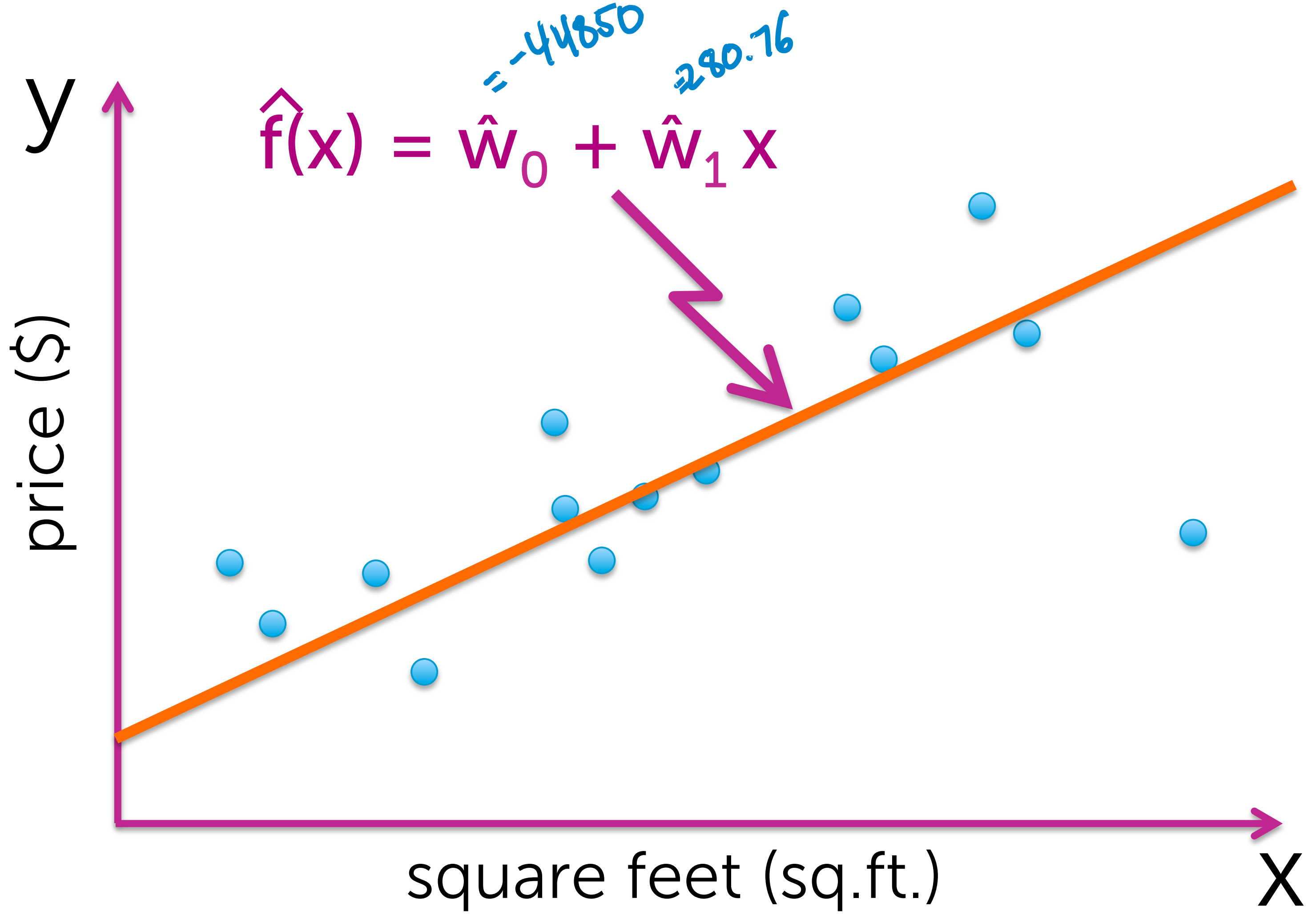
Find “best” line



$$\mathrm{RSS}(w_0,w_1) = \sum_{i=1}^{N} (y_i-[w_0+w_1x_i])^2$$

The fitted line: use + interpretation

Model vs. fitted line



Regression model:

$$y_i = w_0 + w_1 x_i + \varepsilon_i$$

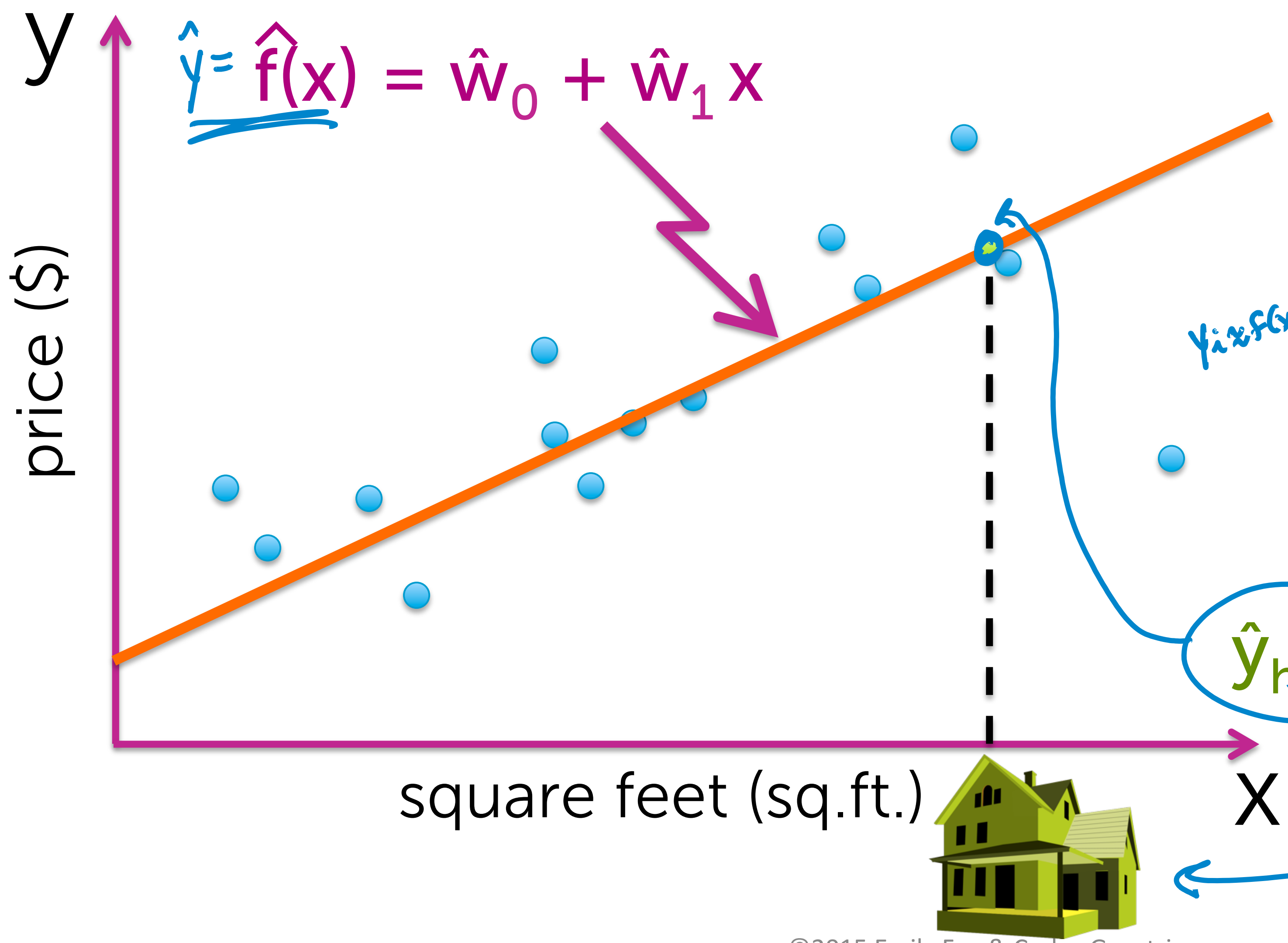
parameters (unknown variables)

Estimated parameters:

$$\hat{w}_0 = -44850, \ \hat{w}_1 = 280.76$$

take actual values

Seller:
Predicting your house price



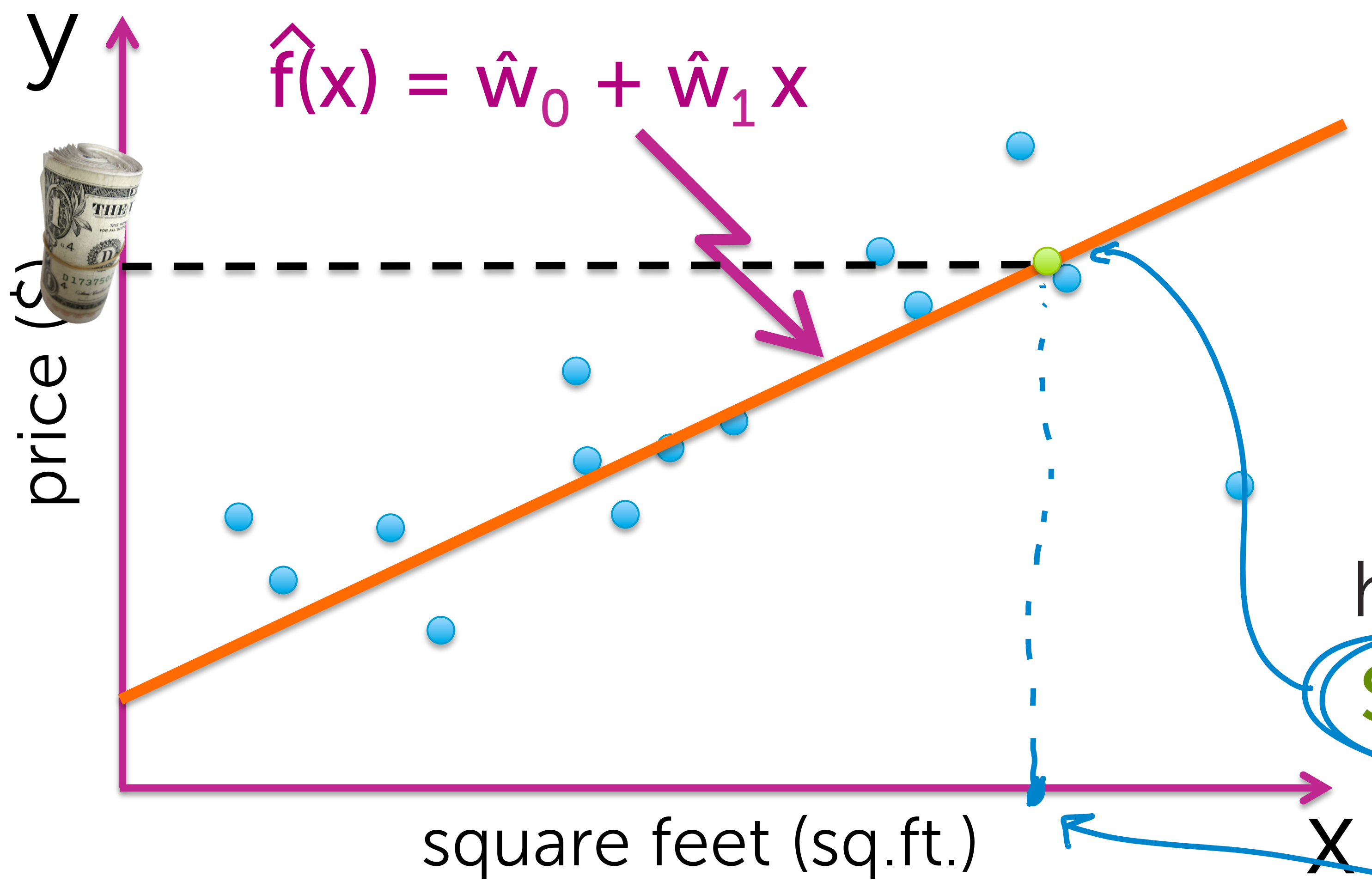
Regression model:

$y_i \approx f(x_i)$: $y_i = w_0 + w_1 x_i + \varepsilon_i$

Best guess of your house price:

$\hat{y}_{house} = \hat{w}_0 + \hat{w}_1 \text{sq.ft.}_{house}$

Buyer:
Predicting size of house



Regression model:

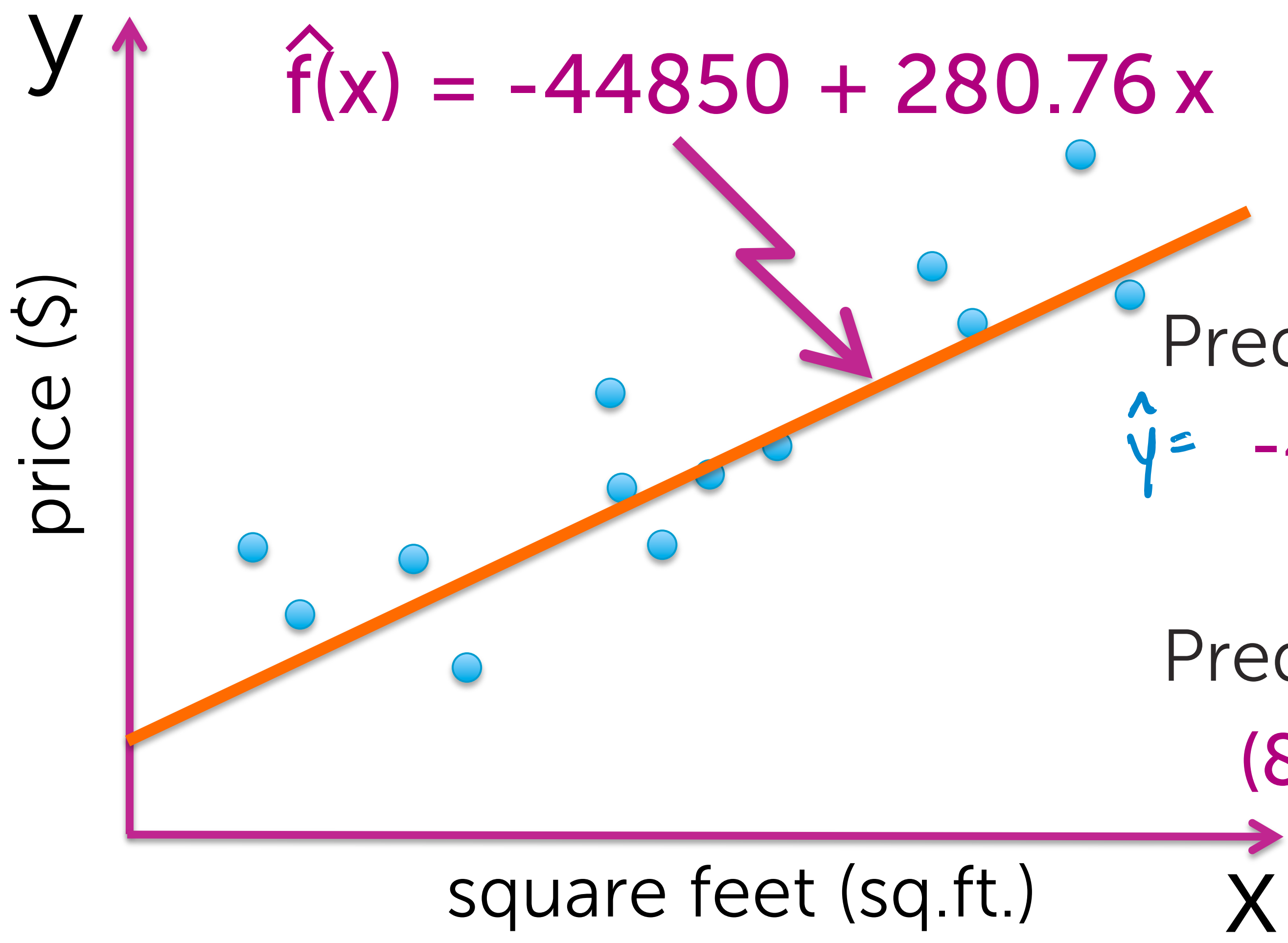
$$y_i = w_0 + w_1 x_i + \varepsilon_i$$

Best guess of size of house you can afford:

$$\$_{\text{in bank}} = \hat{w}_0 + \hat{w}_1 \widehat{\text{sq.ft.}}$$

A concrete example

$$\hat{f}(x) = -44850 + 280.76\,x$$



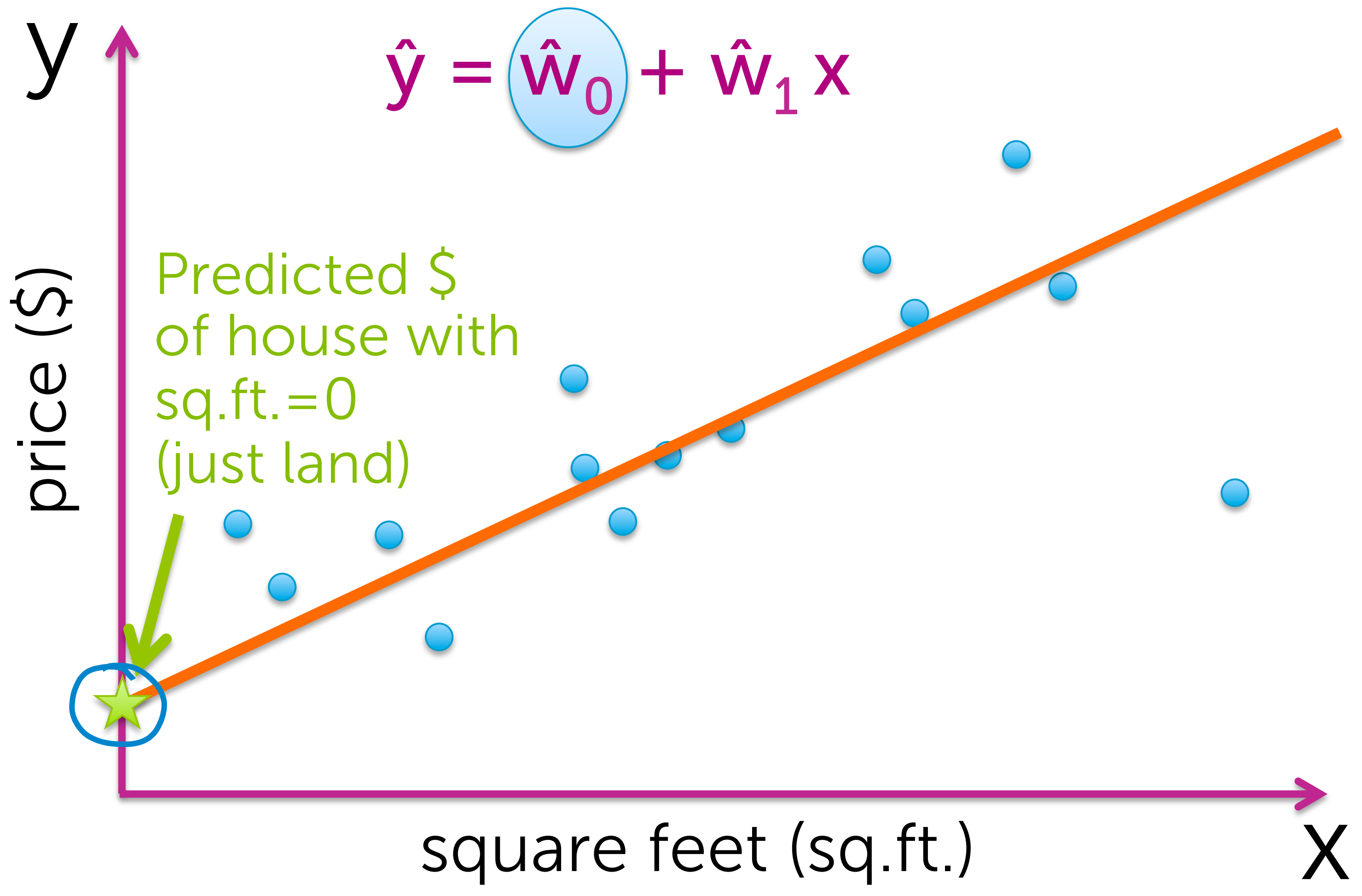
Predict $ of 2,640 sq.ft. house:

$$\hat{y} = -44850 + 280.76 * \underline{2,640}$$
$$= \$696,356$$

Predict sq.ft. of $\underline{\$859,000}$ sale:

$$(859000+44850)/\ 280.76$$
$$= 3,219 \text{ sq.ft.}$$

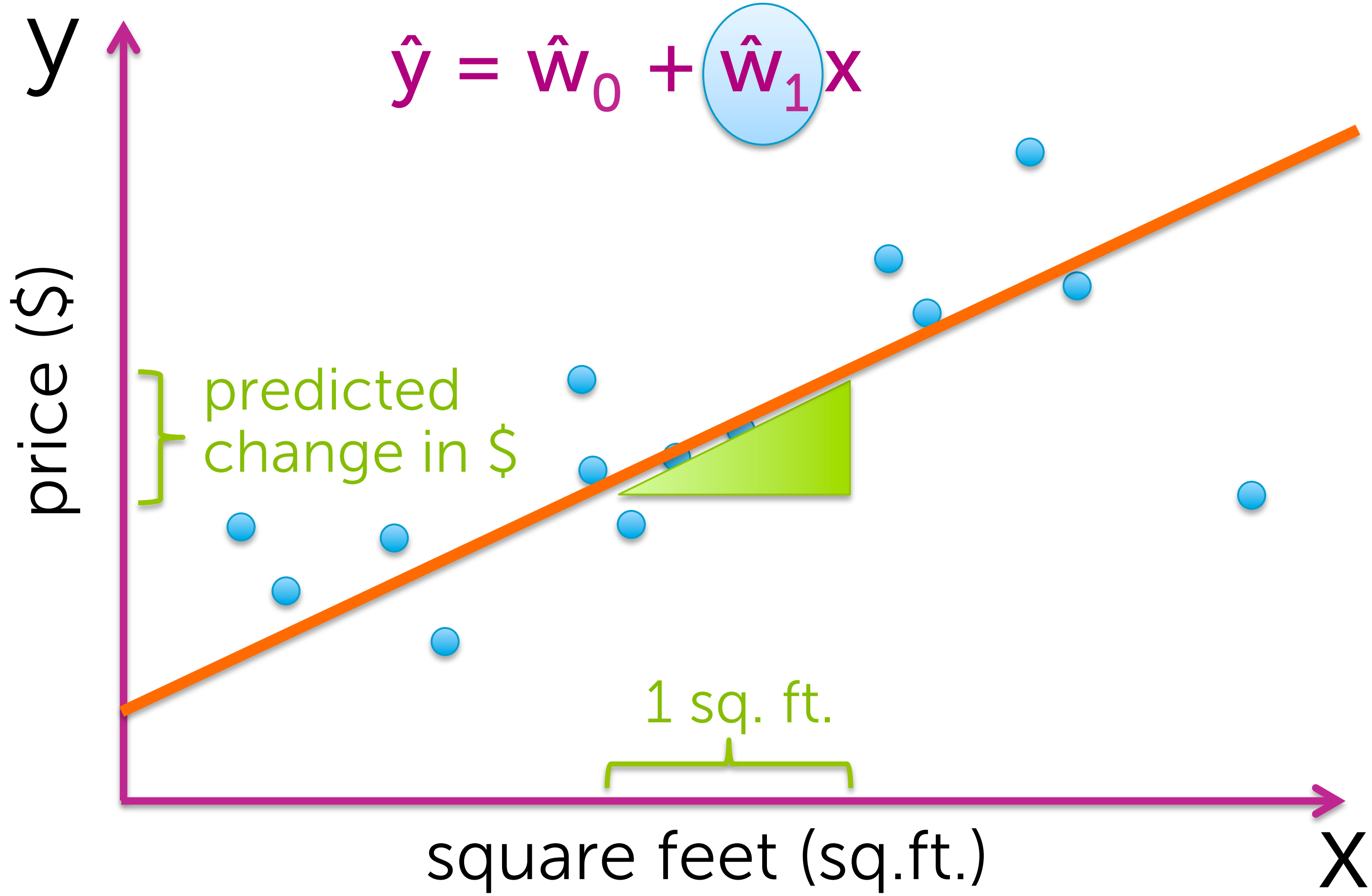
Interpreting the coefficients



$\hat{y} = \hat{w}_0$ when $x=0$

not very meaningful

Interpreting the coefficients



$\hat{\$}_{1001\text{ sq.ft.}} - \hat{\$}_{1000\text{ sq.ft.}}$

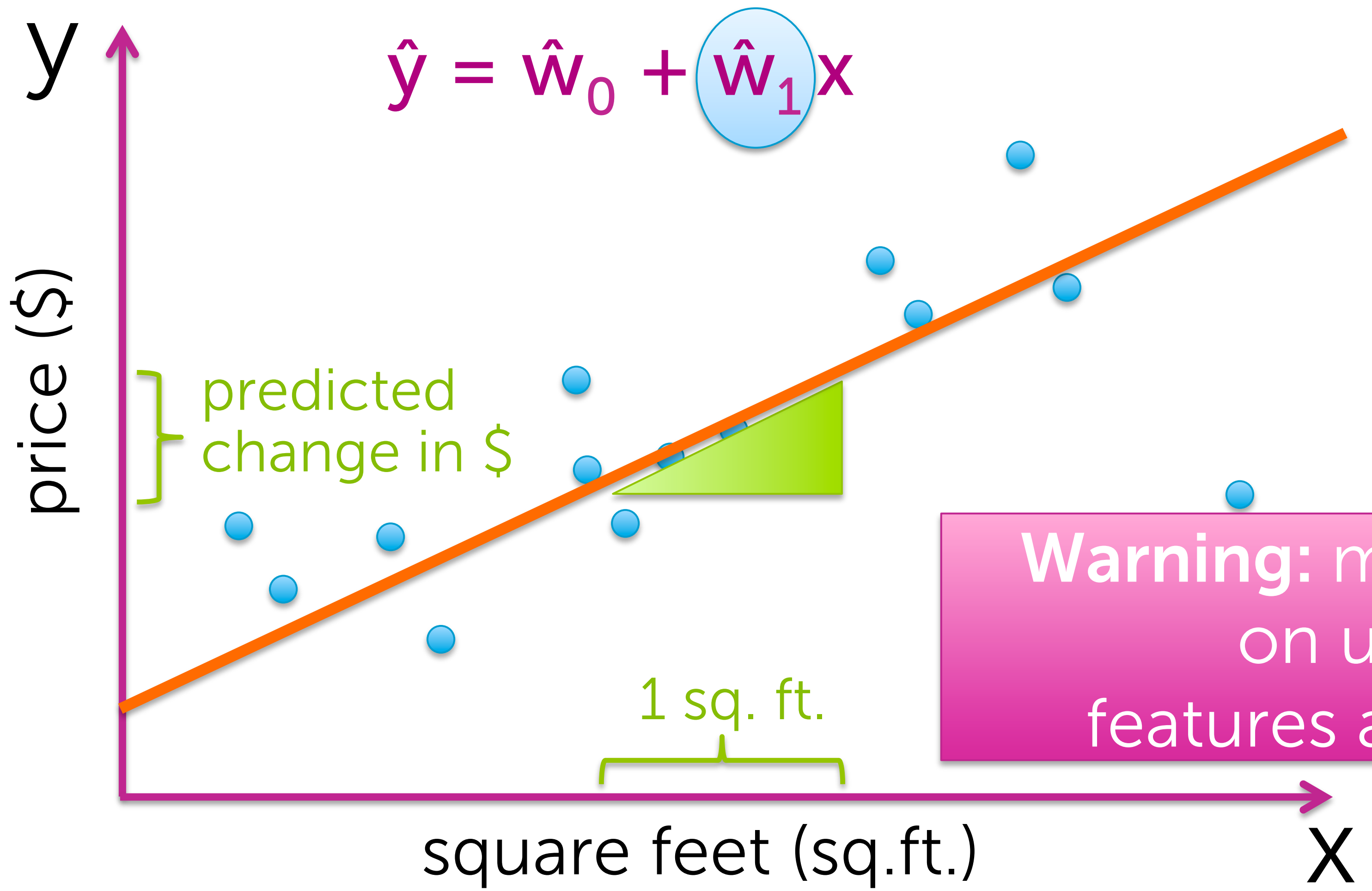
$= \cancel{\hat{w}_0} + \hat{w}_1 \cdot 1001 \text{ sq.ft.}$

$- (\cancel{\hat{w}_0} + \hat{w}_1 1000 \text{ sq.ft.})$

$= \hat{w}_1$

predicted change in the output per unit change in input

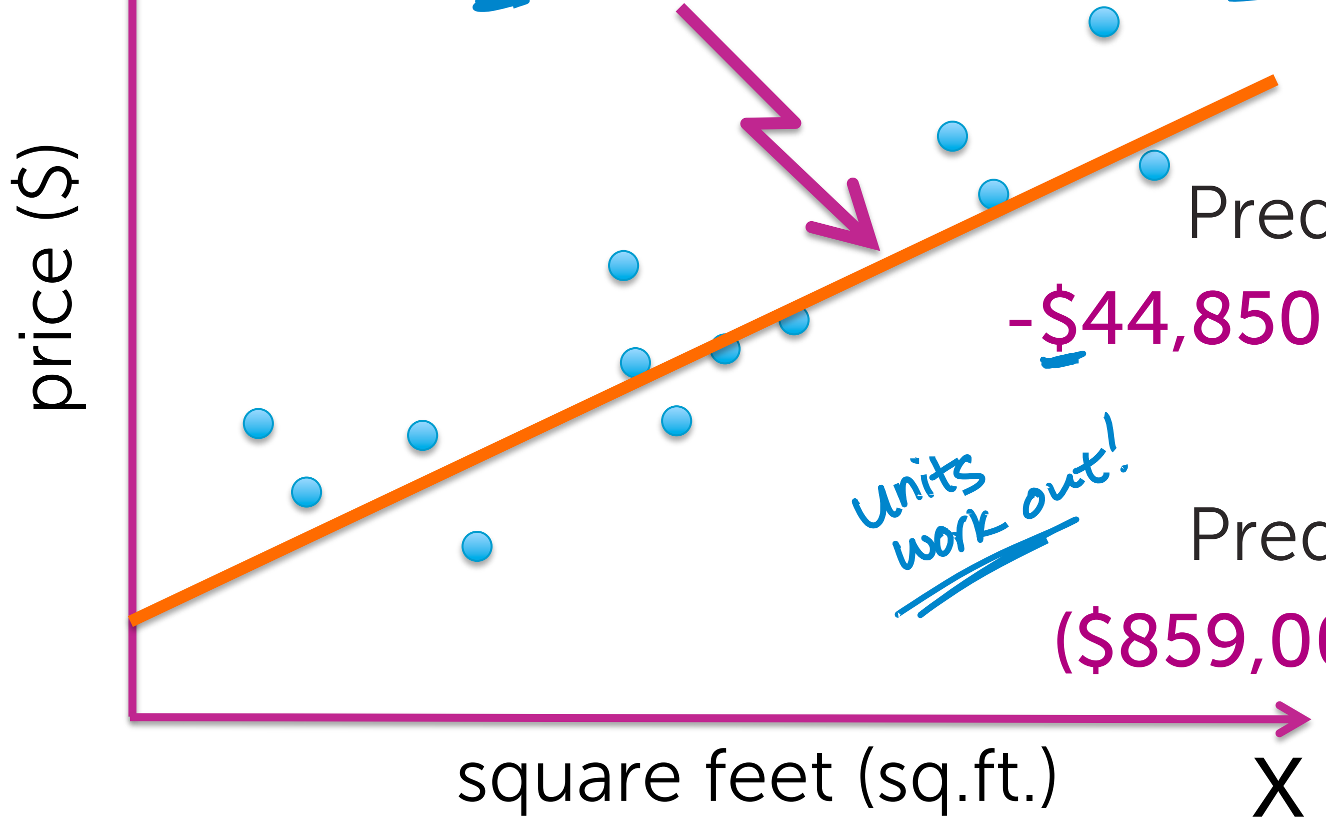
Interpreting the coefficients



Warning: magnitude depends on units of both features and observations

A concrete example

$$\hat{f}(x) = -\$44{,}850 + 280.76\ (\$/\text{sq.ft.})\ x$$



Predict $ of 2,640 sq.ft. house:

$$-\$44{,}850 + 280.76\ (\$/\text{sq.ft.}) * 2{,}640\ \text{sq.ft.}$$

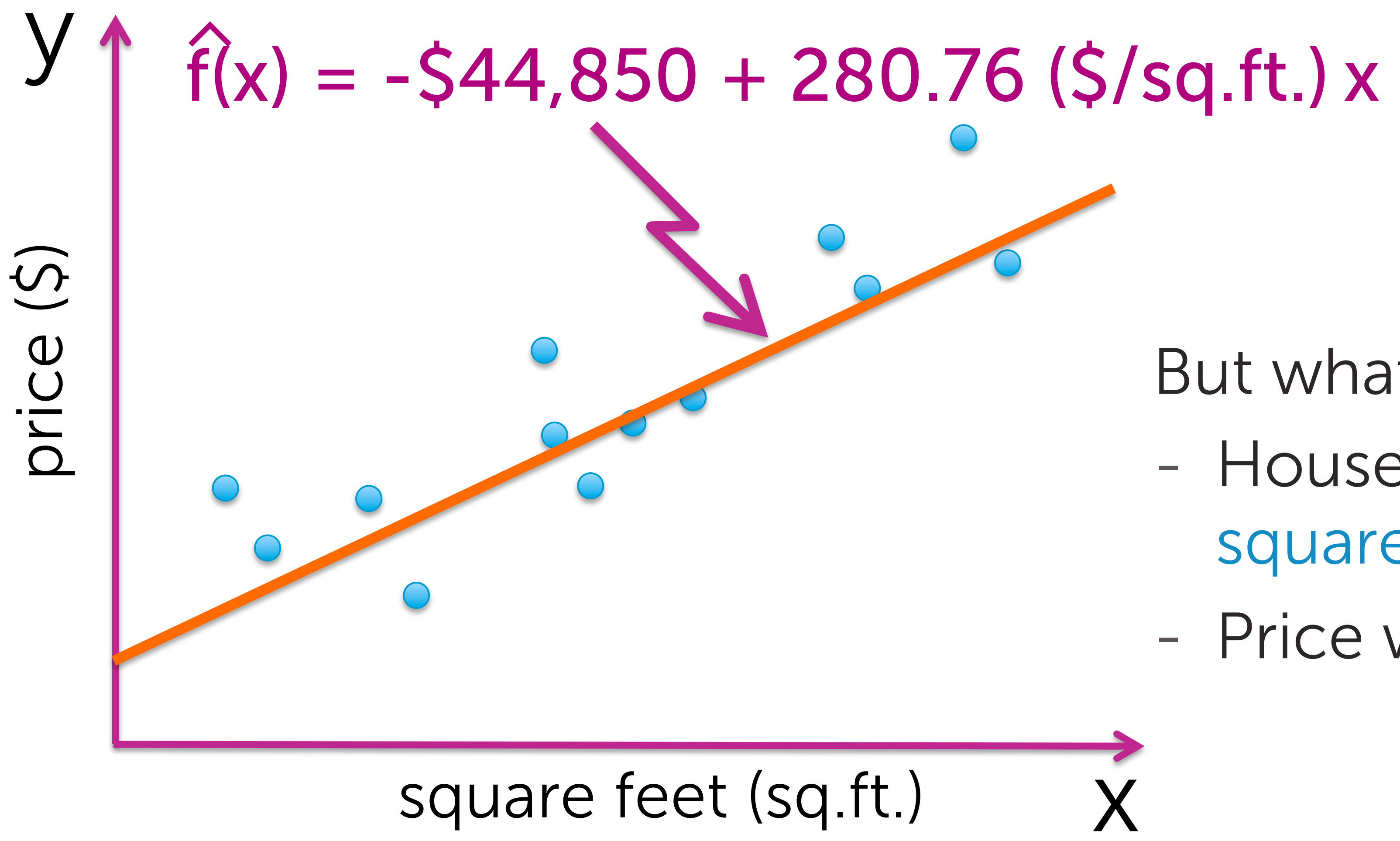
$$= \$696{,}356$$

Predict sq.ft. of $859,000 sale:

$$(\$859{,}000+\$44{,}850)/\ 280.76\ (\$/\text{sq.ft.})$$

$$= 3{,}219\ \text{sq.ft.}$$

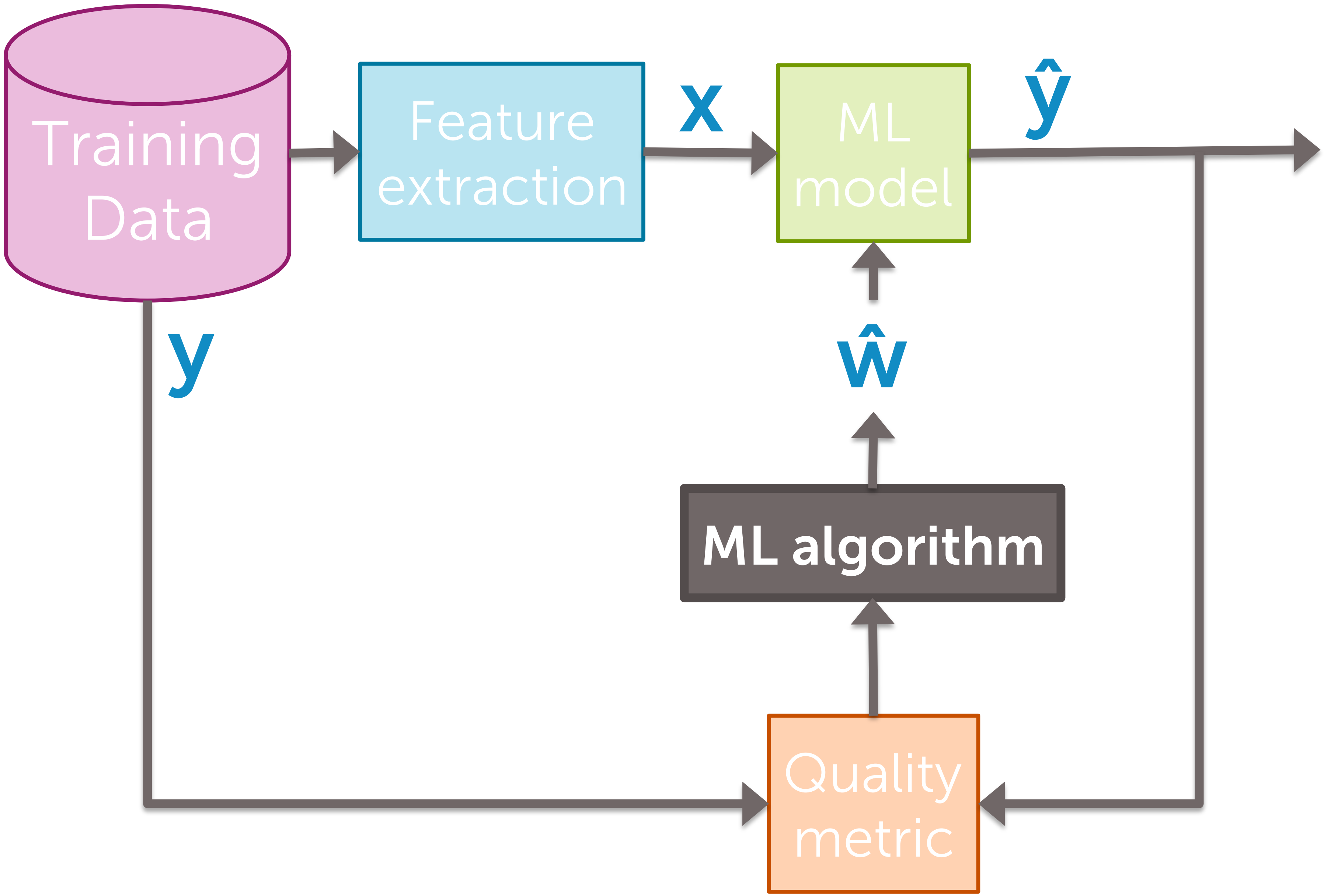
A concrete example



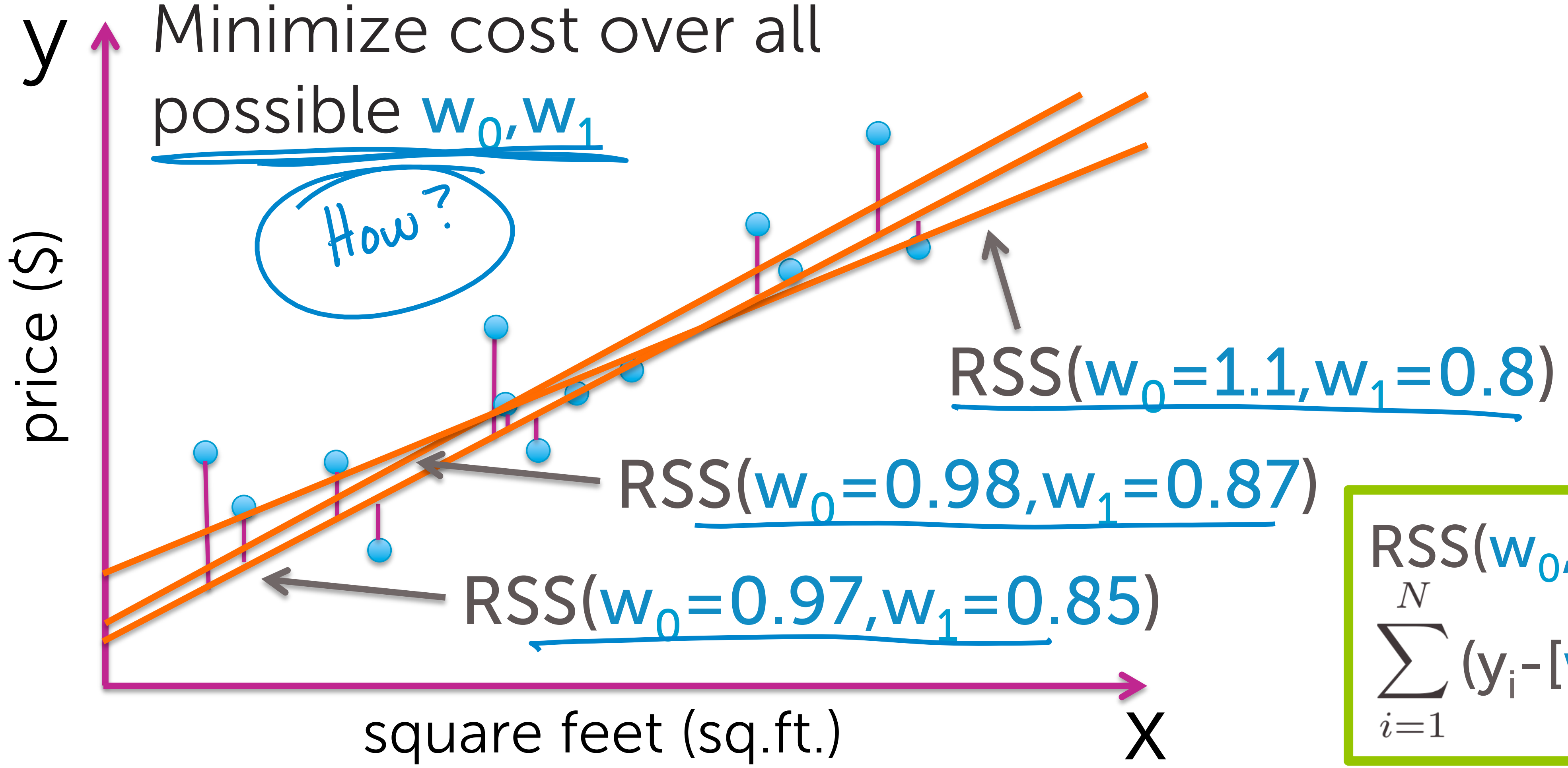
But what if:

- House was measured in square meters?
- Price was measured in RMB?

Algorithms for fitting the model


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

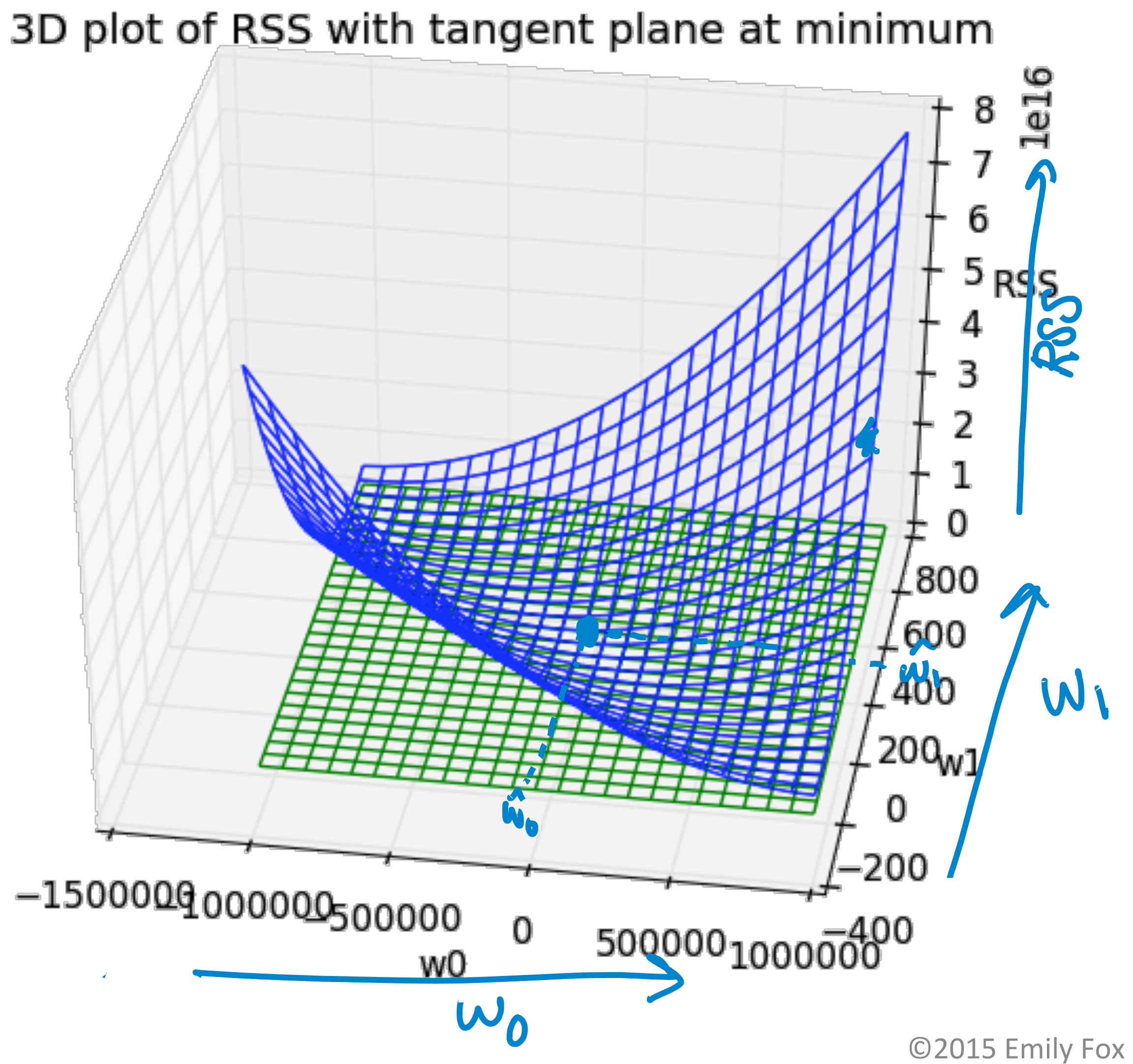
Find "best" line



Recall:

$$RSS(w_0,w_1) = \sum_{i=1}^{N}(y_i-[w_0+w_1x_i])^2$$

Minimizing the cost



Minimize function over all possible w_0, w_1

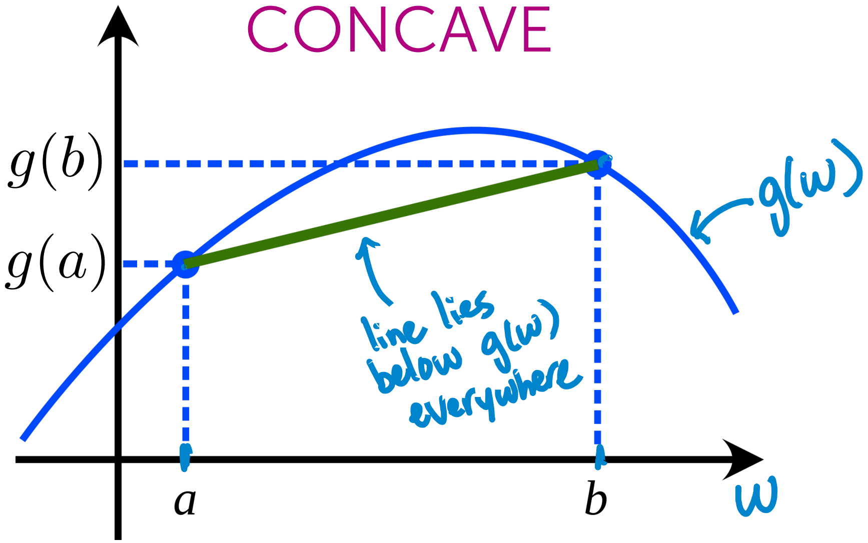
$$\boxed{\min_{w_0,w_1}} \underbrace{\sum_{i=1}^{N} (y_i - [w_0 + w_1 x_i])^2}$$

RSS(w_0,w_1) is a function of 2 variables $= g(w_0, w_1)$

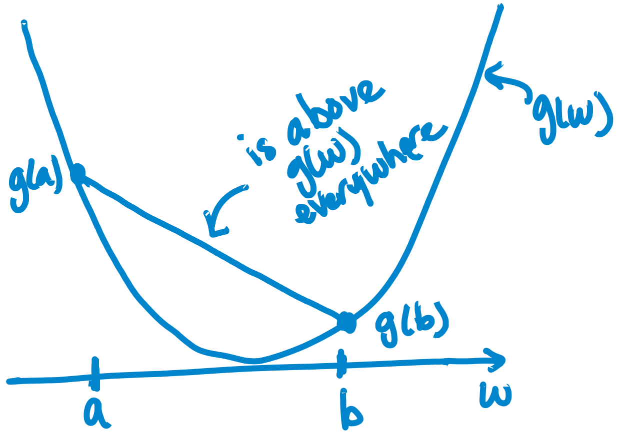
An aside on optimization

Convex/concave functions

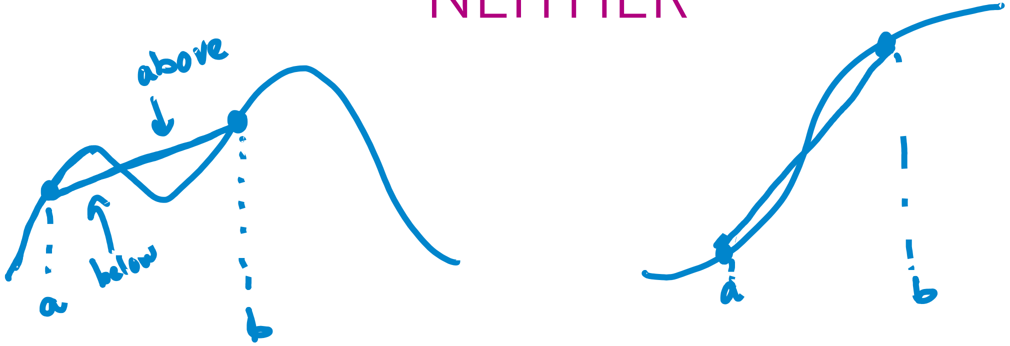
CONCAVE



CONVEX

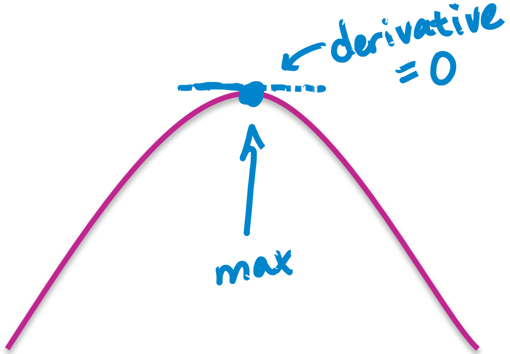


NEITHER



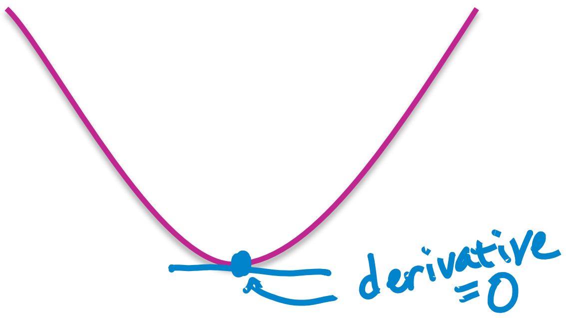
Finding the max or min analytically

CONCAVE



$\max_{w} g(w)$

CONVEX



$\min_{w} g(w)$

NEITHER

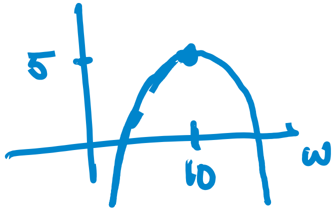
Example:

g(w) = 5-(w-10)²

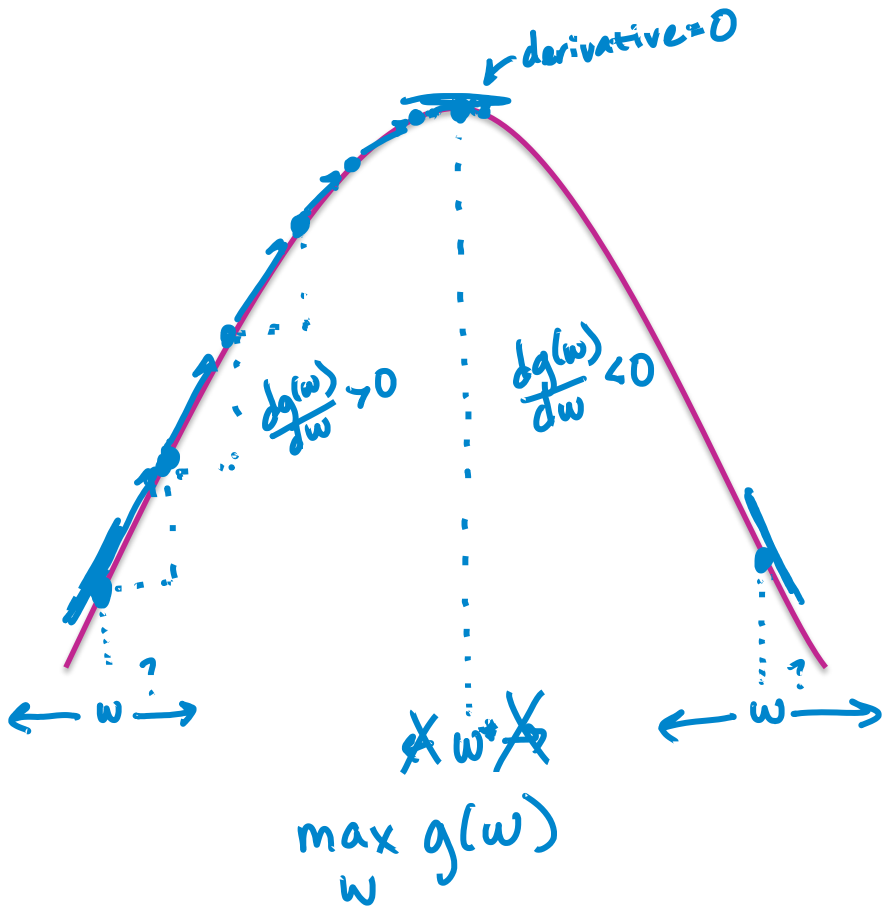
$$\frac{dg(w)}{dw} = 0 - 2(w-10)^1 \cdot 1$$
$$= -2w + 20$$

set derivate = 0:

$$-2w+20=0$$
$$w = 10$$



Finding the max via hill climbing



How do we know whether to move w to right or left? (inc. or dec. the value of w?)

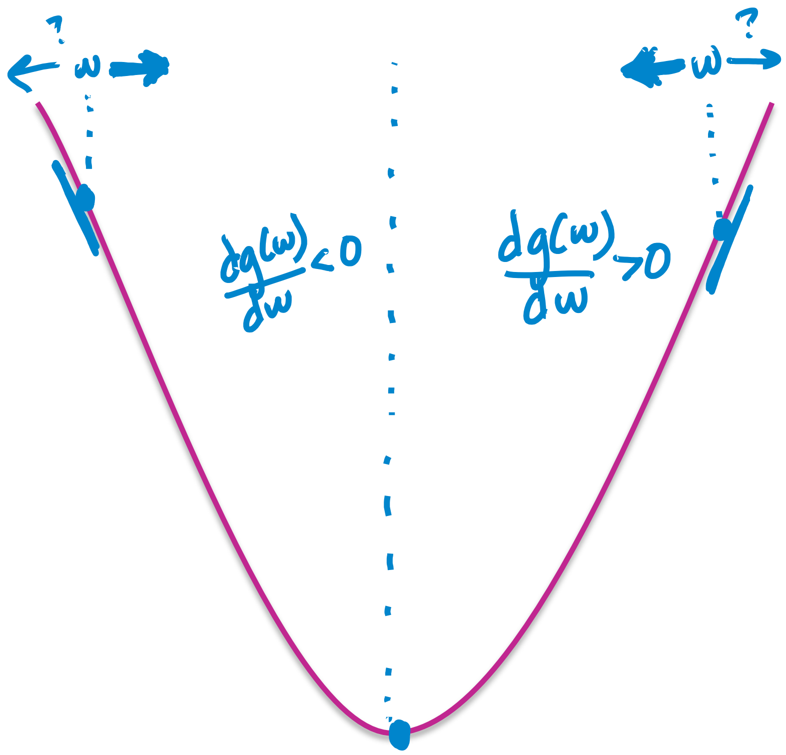
while not converged

$$w^{(t+1)} \leftarrow w^{(t)} + \eta \frac{dg(w)}{dw}$$

iteration t

stepsize

Finding the min via hill descent



$\min_w g(w)$

when derivative is positive, we want to decrease w
and when derivative is negative, we want to increase w

Algorithm:

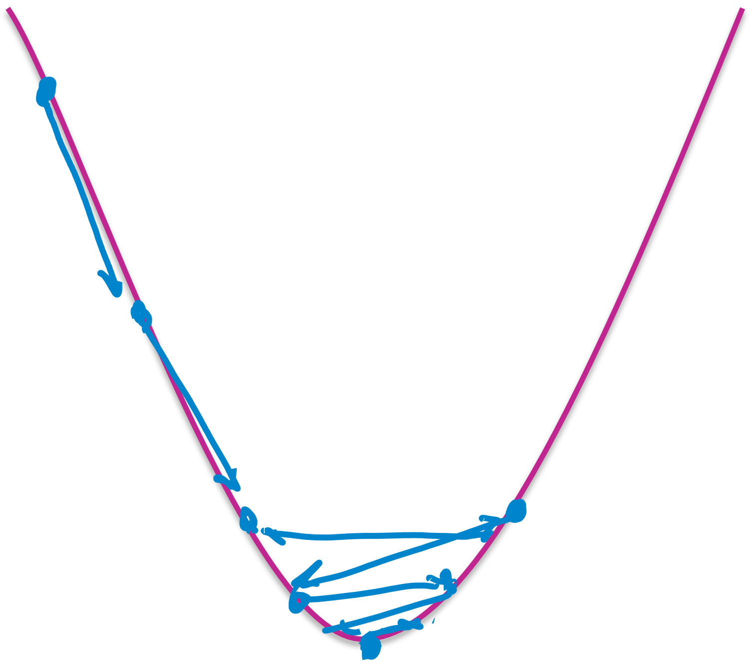
while not converged

$$w^{(t+1)} \leftarrow w^{(t)} - \eta \left.\frac{dg}{dw}\right|_{w^{(t)}}$$

Choosing the stepsize—

Fixed stepsize

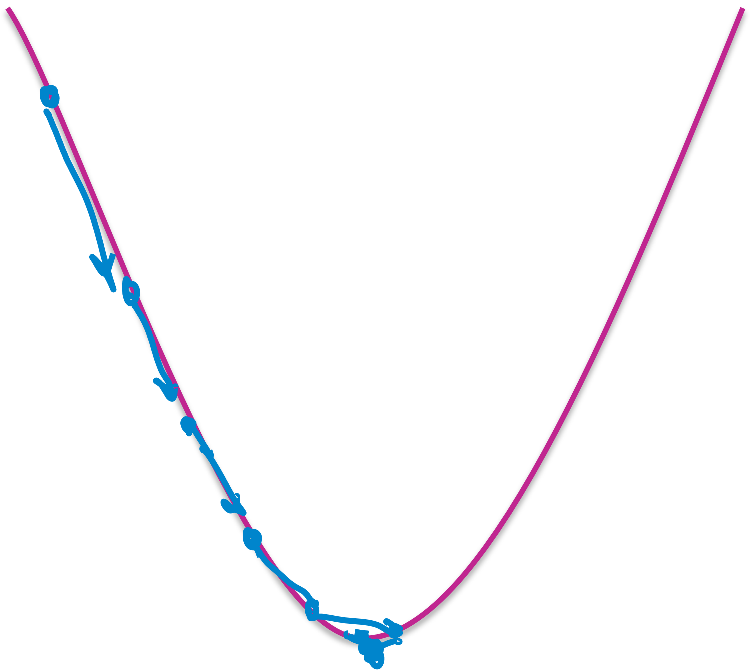
η



$\eta = 0.1$

Choosing the stepsize—
Decreasing stepsize

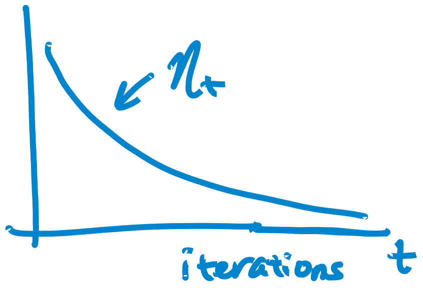
or stepsize schedule



Common choices:

$$\eta_t = \frac{\alpha}{t}$$

$$\eta_t = \frac{\alpha}{\sqrt{t}}$$



Convergence criteria

For convex functions,
optimum occurs when

$$\frac{dg(w)}{dw} = 0$$

In practice, stop when

$$\left|\frac{dg(w)}{dw}\right| < \epsilon$$

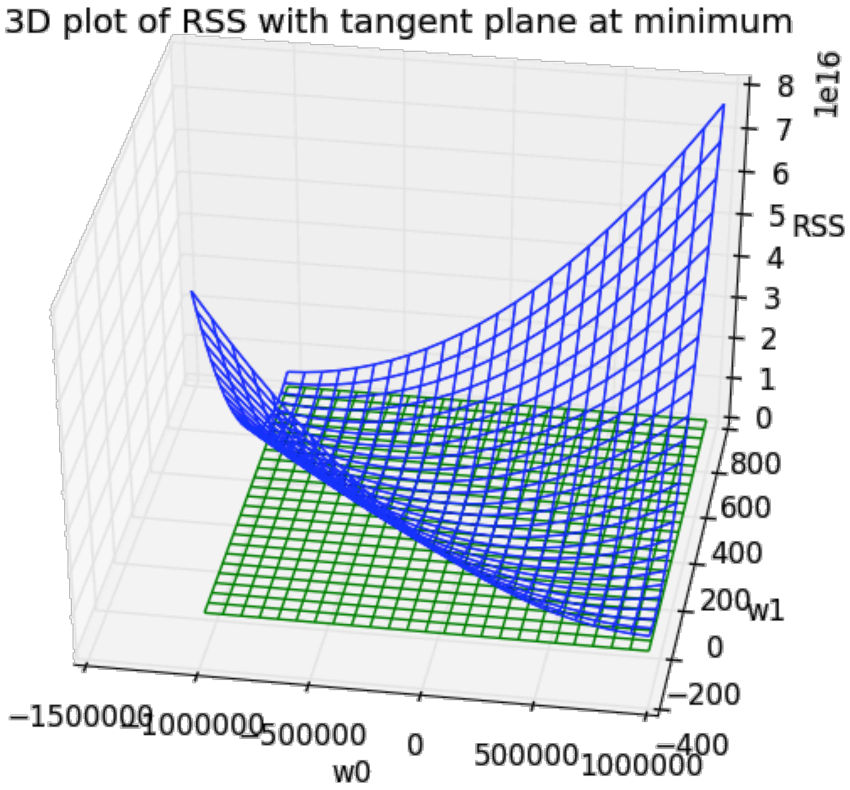
threshold to be set

Algorithm:

while not converged

$$w^{(t+1)} \leftarrow w^{(t)} - \eta \left.\frac{dg}{dw}\right|_{w^{(t)}}$$

Moving to multiple dimensions: Gradients



$$\nabla g(\mathbf{w}) = \begin{bmatrix} \frac{\partial g}{\partial w_0} \\ \frac{\partial g}{\partial w_1} \\ \vdots \\ \frac{\partial g}{\partial w_p} \end{bmatrix}$$

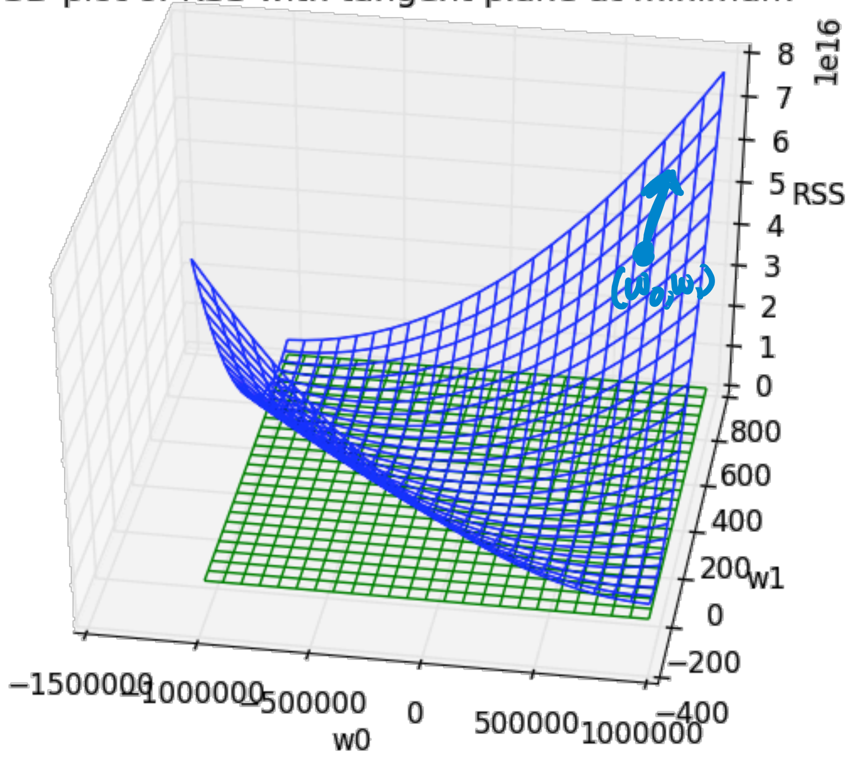
gradient

$[w_0, w_1, \ldots, w_p]$

(p+1)-dimensional vector

partial derivative is like a derivate with respect to w_1 treating all other variables as constants

Gradient example



$$g(\mathbf{w}) = 5w_0 + 10w_0w_1 + 2w_1^2$$

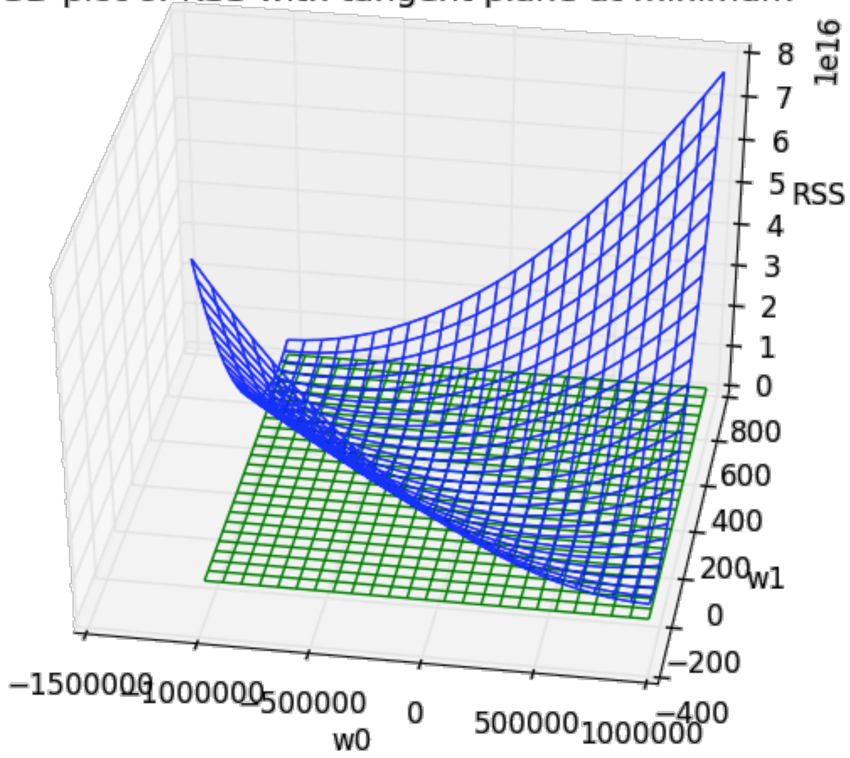
$$\frac{\partial g}{\partial w_0} = 5 + 10w_1$$

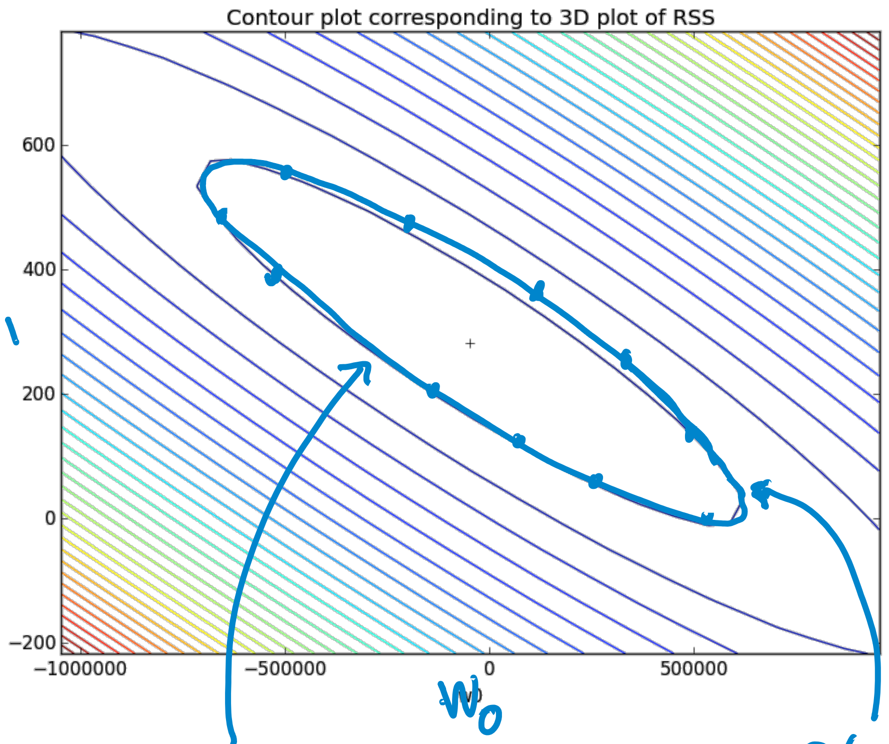
$$\frac{\partial g}{\partial w_1} = 10w_0 + 4w_1$$

$$\nabla g(\mathbf{w}) = \begin{bmatrix} 5 + 10w_1 \\ 10w_0 + 4w_1 \end{bmatrix}$$

Contour plots

bird's eye view

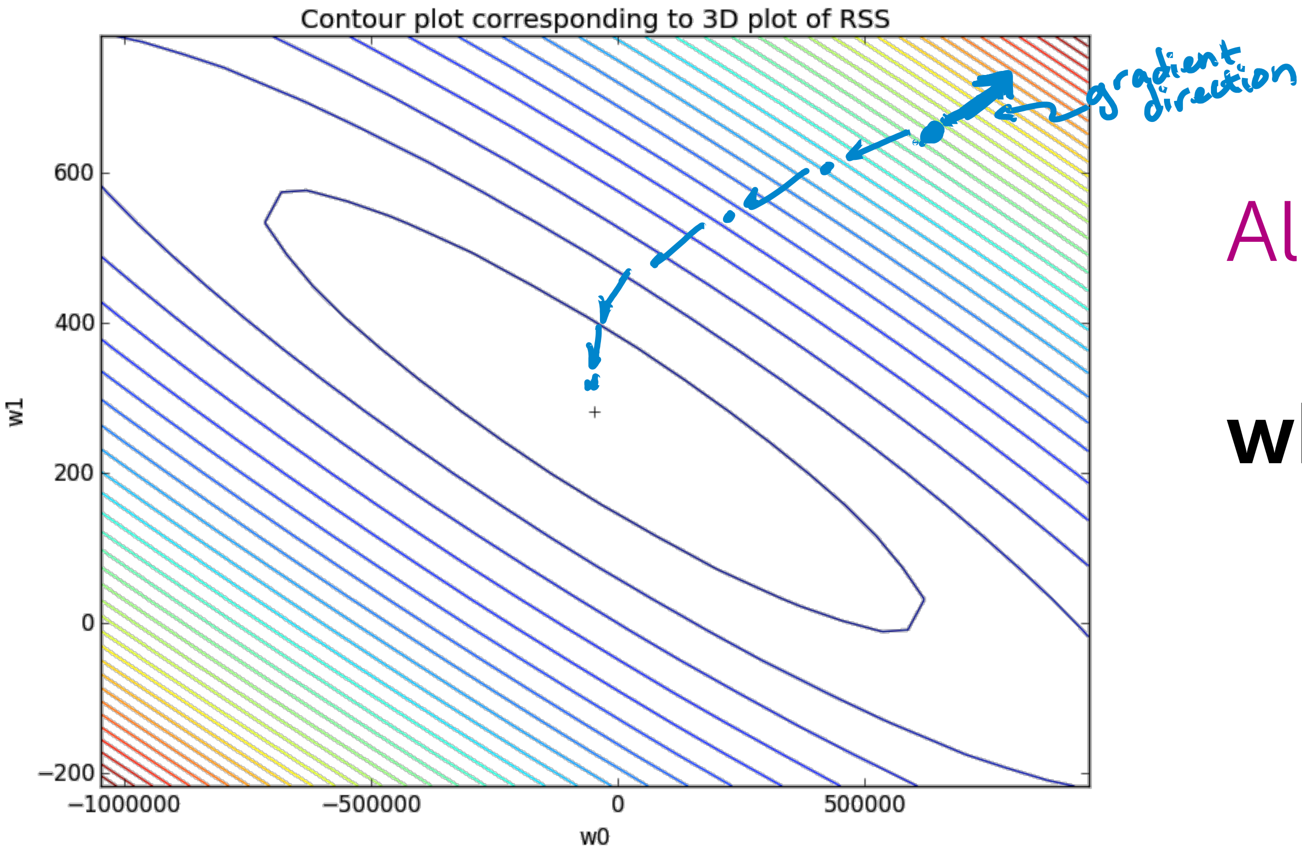




a slice of the 3D surface

$g(w_0, w_1)$

Gradient descent



Algorithm:

while not converged

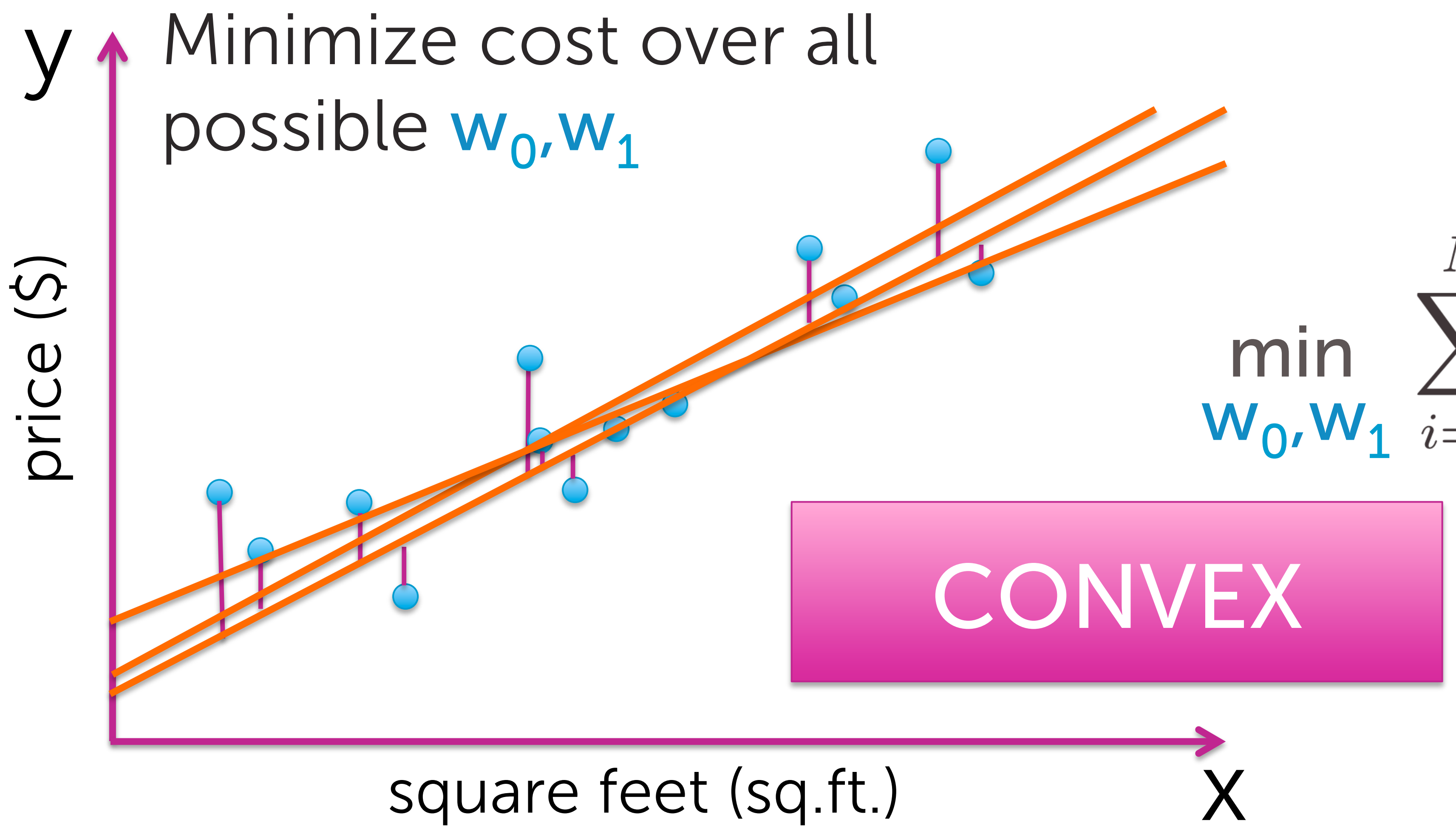
$$w^{(t+1)} \leftarrow w^{(t)} - \eta \nabla g(w^{(t)})$$

$$\begin{bmatrix} \vdots \end{bmatrix} \leftarrow \begin{bmatrix} \vdots \end{bmatrix} - \eta \begin{bmatrix} \vdots \end{bmatrix}$$

$\boxed{\text{Convergence: } \|\nabla g(w)\| < \epsilon}$

Finding the least squares line

Find "best" line



$$\min_{w_0,w_1} \sum_{i=1}^{N} (y_i - [w_0 + w_1 x_i])^2$$

$\Rightarrow$ solution is unique

+ gradient descent alg. will converge to minimum

Compute the gradient

$$RSS(w_0,w_1) = \sum_{i=1}^{N}(y_i-[w_0+w_1x_i])^2$$

Aside:

$$\frac{d}{dw}\sum_{i=1}^{N} g_i(w) = \frac{d}{dw}\left(g_1(w) + g_2(w) + \ldots g_N(w)\right)$$

$$= \frac{d}{dw}g_1(w) + \frac{d}{dw}g_2(w) + \ldots \frac{d}{dw}g_N(w)$$

$$= \sum_{i=1}^{N}\frac{d}{dw}g_i(w)$$

In our case

$g_i(w) = (y_i - [w_0+w_1x_i])^2$

$$\frac{\partial RSS(w)}{\partial w_0} = \sum_{i=1}^{N}\frac{\partial}{\partial w_0}(y_i-[w_0+w_1x_i])^2$$

same for w_1

Compute the gradient

$$RSS(w_0,w_1) = \sum_{i=1}^{N}(y_i-[w_0+w_1x_i])^2$$

Taking the derivative w.r.t. w_0

$$\sum_{i=1}^{N} 2(y_i-[w_0+w_1x_i])^1 \cdot (-1)$$

$$= -2\sum_{i=1}^{N}(y_i-[w_0+w_1x_i])$$

Compute the gradient

$$RSS(w_0,w_1) = \sum_{i=1}^{N} (y_i-[w_0+w_1x_i])^2$$

Taking the derivative w.r.t. w_1

$$\sum_{i=1}^{N} 2(\underline{y_i-[w_0+w_1x_i]})^1 \cdot (-x_i)$$

$$= -2\sum_{i=1}^{N} (y_i-[w_0+w_1x_i])\underline{x_i}$$

Compute the gradient

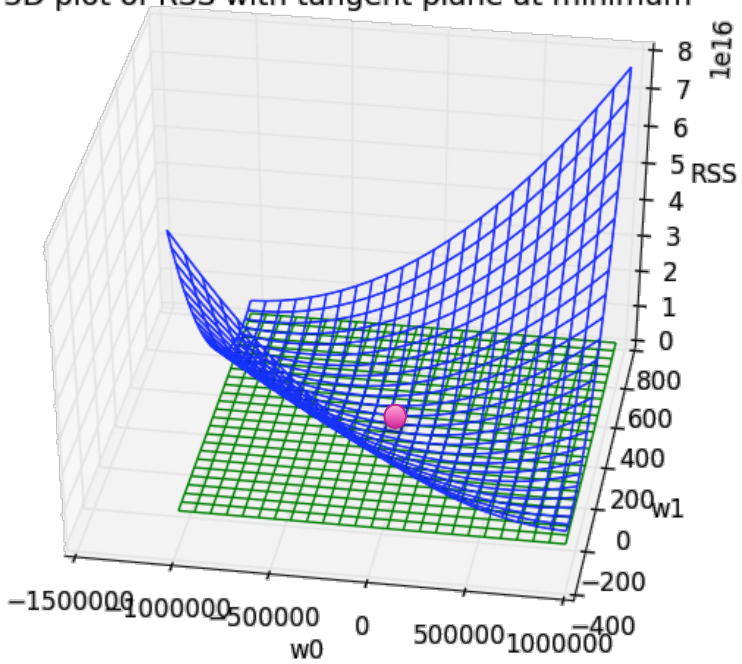
$$RSS(w_0,w_1) = \sum_{i=1}^{N}(y_i-[w_0+w_1x_i])^2$$

Putting it together:

$$\nabla RSS(w_0,w_1) = \begin{bmatrix} -2\sum_{i=1}^{N}[y_i - (w_0+w_1x_i)] \\ -2\sum_{i=1}^{N}[y_i - (w_0+w_1x_i)]x_i \end{bmatrix}$$

Approach 1: Set gradient = 0

$$\nabla RSS(w_0,w_1) = \begin{bmatrix} -2\sum_{i=1}^{N}[y_i - (w_0+w_1x_i)] \\ -2\sum_{i=1}^{N}[y_i - (w_0+w_1x_i)]x_i \end{bmatrix}$$



top term:

$$\hat{w}_0 = \frac{\sum_{i=1}^{N} y_i}{N} - \hat{w}_1 \frac{\sum_{i=1}^{N} x_i}{N}$$

average house sales price

estimate of the slope

average sq.ft.

bottom term:

$$\sum y_i x_i - \hat{w}_0 \sum x_i - \hat{w}_1 \sum x_i^2 = 0$$

plug in

$$\hat{w}_1 = \frac{\sum y_i x_i - \frac{\sum y_i \sum x_i}{N}}{\sum x_i^2 - \frac{\sum x_i \sum x_i}{N}}$$

Note:

$\sum_{i=1}^{N} y_i$

$\sum_{i=1}^{N} x_i$

$\sum_{i=1}^{N} y_i x_i$

$\sum_{i=1}^{N} x_i^2$

Approach 2: Gradient descent

Interpreting the gradient:

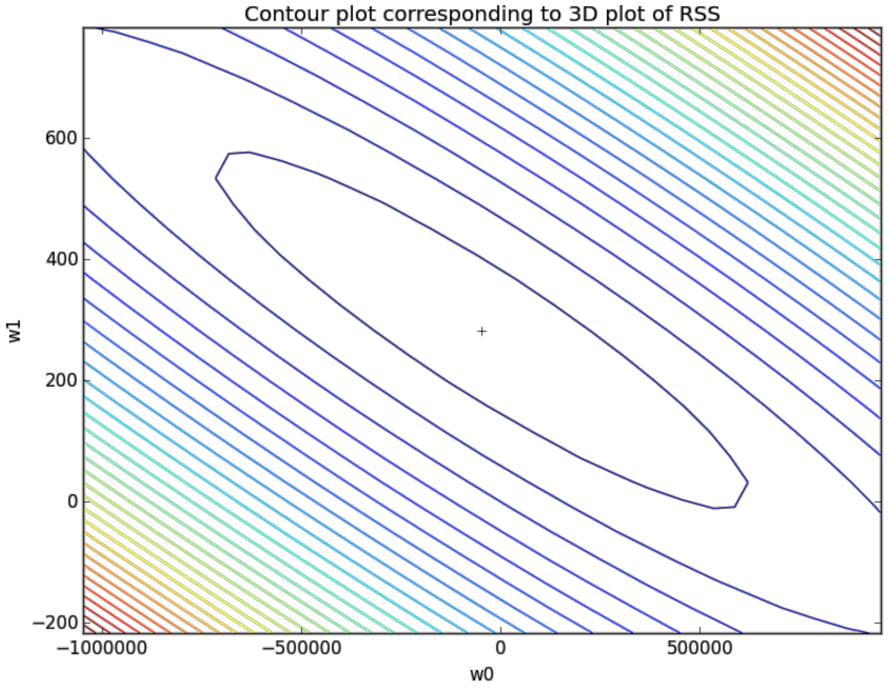
actual house sales observation

predicted value $\hat{y}_i(w_0, w_1)$

$$\nabla RSS(w_0,w_1) = \begin{bmatrix} -2\sum_{i=1}^{N}[y_i - \underline{(w_0+w_1x_i)}] \\ -2\sum_{i=1}^{N}[y_i - (w_0+w_1x_i)]x_i \end{bmatrix} = \begin{bmatrix} -2\sum_{i=1}^{N}[y_i - \hat{y}_i(w_0,w_1)] \\ -2\sum_{i=1}^{N}[y_i - \hat{y}_i(w_0,w_1)]x_i \end{bmatrix}$$

Approach 2: Gradient descent

$$\nabla RSS(w_0,w_1) = \begin{bmatrix} -2\sum_{i=1}^{N}[y_i - \hat{y}_i(w_0,w_1)] \\ -2\sum_{i=1}^{N}[y_i - \hat{y}_i(w_0,w_1)]x_i \end{bmatrix}$$



while not converged

$(-2)\cdot(-\eta)$

$$\begin{bmatrix} w_0^{(t+1)} \\ w_1^{(t+1)} \end{bmatrix} \leftarrow \begin{bmatrix} w_0^{(t)} \\ w_1^{(t)} \end{bmatrix} + 2\eta \begin{bmatrix} \sum_{i=1}^{N}[y_i - \hat{y}_i(w_0^{(t)},w_1^{(t)})] \\ \sum_{i=1}^{N}[y_i - \hat{y}_i(w_0^{(t)},w_1^{(t)})]x_i \end{bmatrix}$$

If overall, underpredicting $\hat{y}_i$, then $\sum[y_i - \hat{y}_i]$ is positive

$\rightarrow$ w_0 is going to increase

similar intuition for w_1, but multiply by x_i

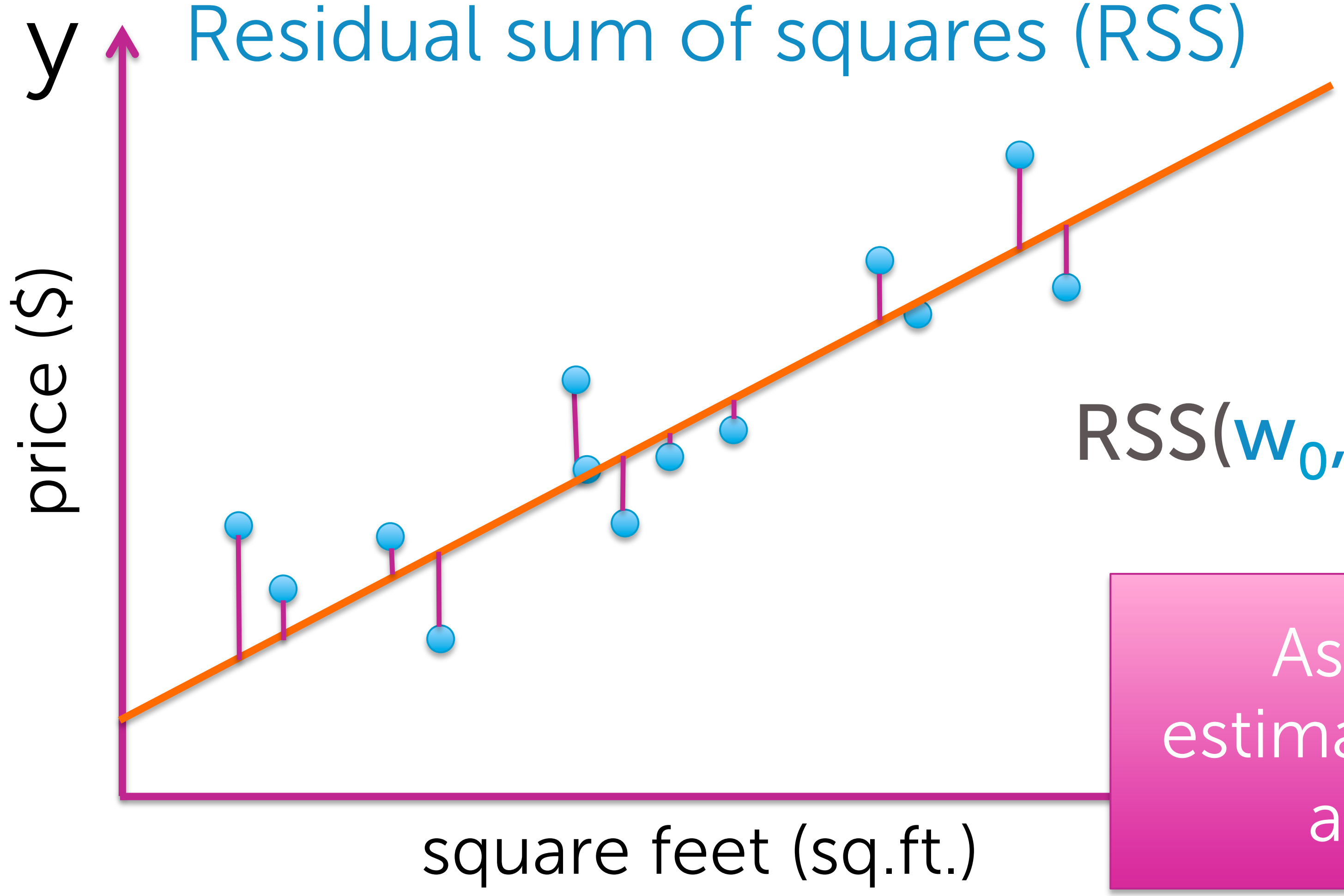
Comparing the approaches

- For most ML problems, cannot solve gradient = 0
- Even if solving gradient = 0 is feasible, gradient descent can be more efficient
- Gradient descent relies on choosing stepsize and convergence criteria

Influence of high leverage points

Asymmetric errors

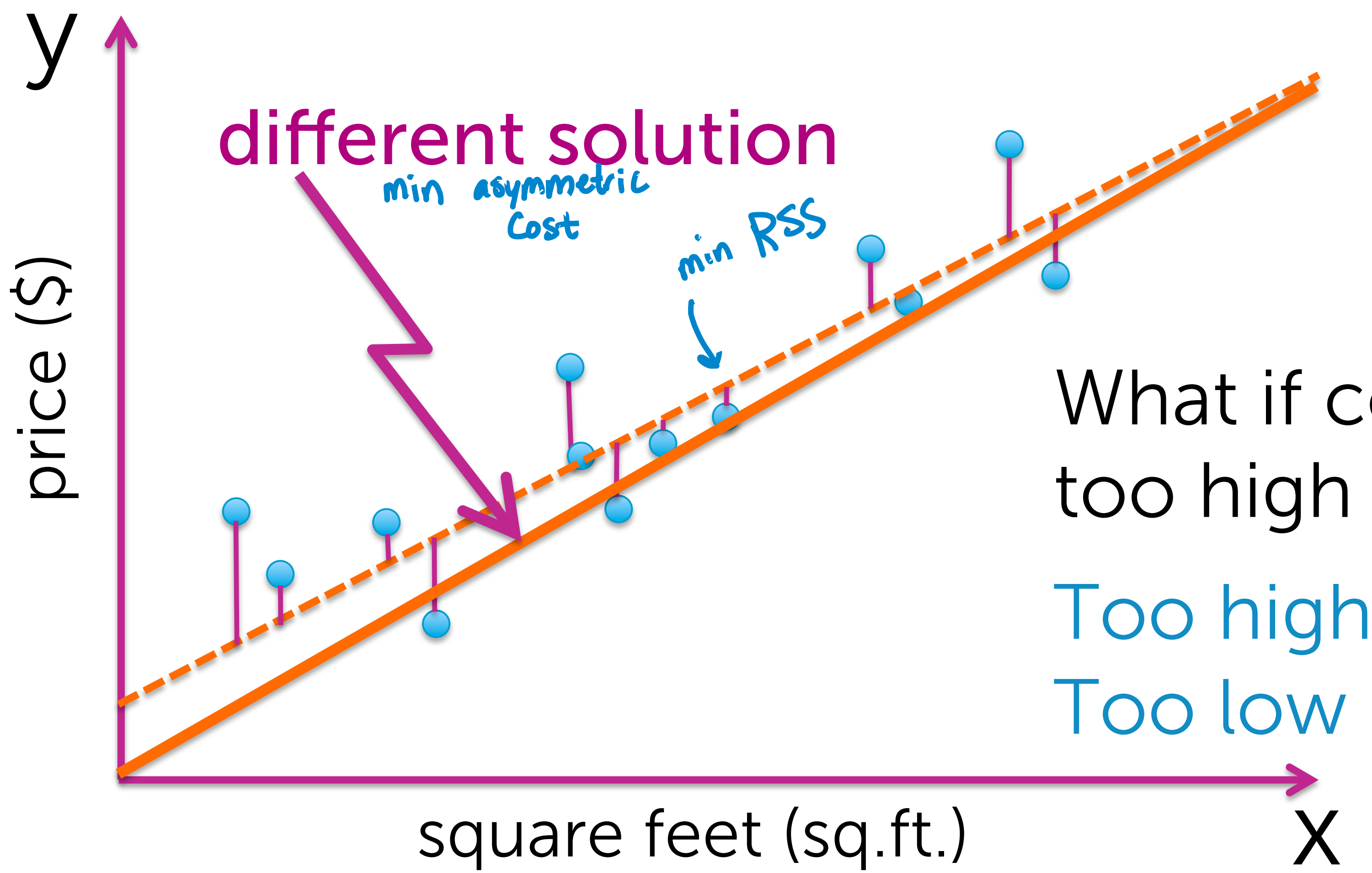
Symmetric cost functions



$$RSS(w_0,w_1) = \sum_{i=1}^{N}(y_i-[w_0+w_1x_i])^2$$

Assumes cost of over-estimating sales price is same as under-estimating

Asymmetric cost functions



What if cost of listing house too high has **bigger cost**?

Too high → no offers ($=0)
Too low → offers for lower $

Summary for simple linear regression

What you can do now…

- Describe the input (features) and output (real-valued predictions) of a regression model
- Calculate a goodness-of-fit metric (e.g., RSS)
- Estimate model parameters to minimize RSS using gradient descent
- Interpret estimated model parameters
- Exploit the estimated model to form predictions
- Discuss the possible influence of high leverage points
- Describe intuitively how fitted line might change when assuming different goodness-of-fit metrics