# Derivatives | Gradients | Gradient Descent

**What Derivatives tell us?**

Derivatives tells us if a function is increasing or decreasing (curve is going upward or downward). If a function is increasing then its derivative is positive and if the function is decreasing then the derivative is negative.

**What Gradient tells us?**

Gradient tells us the steepness of a function. How fast the function is increasing or decreasing.

**What is Gradient Descent?**

It’s an approach to find the optimal value efficiently. It takes bigger steps where the result is far from the optimal and takes smaller steps when it closes to the optimal value. In order to achieve that it uses "derivative".

A parameter called "learning rate" defines how big those starting steps should be.

**What is Stochastic Gradient Descent**

For functions where lot of parameters are involved gradient descent approach can take long time. In those cases, stochastic gradient descent optimizes the process by randomly selecting subset of the data at every step rather than the full dataset.

# Regression

1. Simple Regression:
   1. Goodness-of-fit matric:

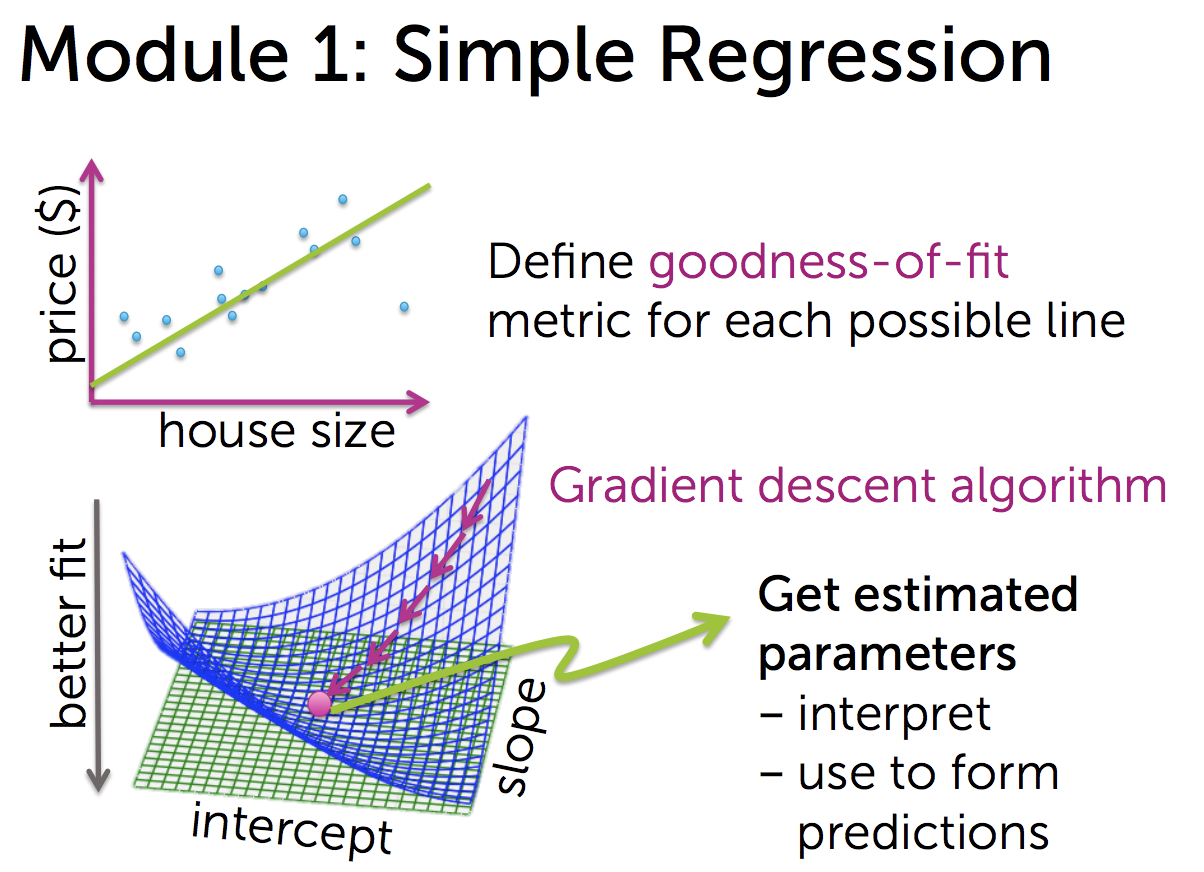
(find the best fitted line)

Define a matric (goodness-of-fit) that allows us to say that some lines fit better than other lines.

* 1. Gradient descent algorithm:

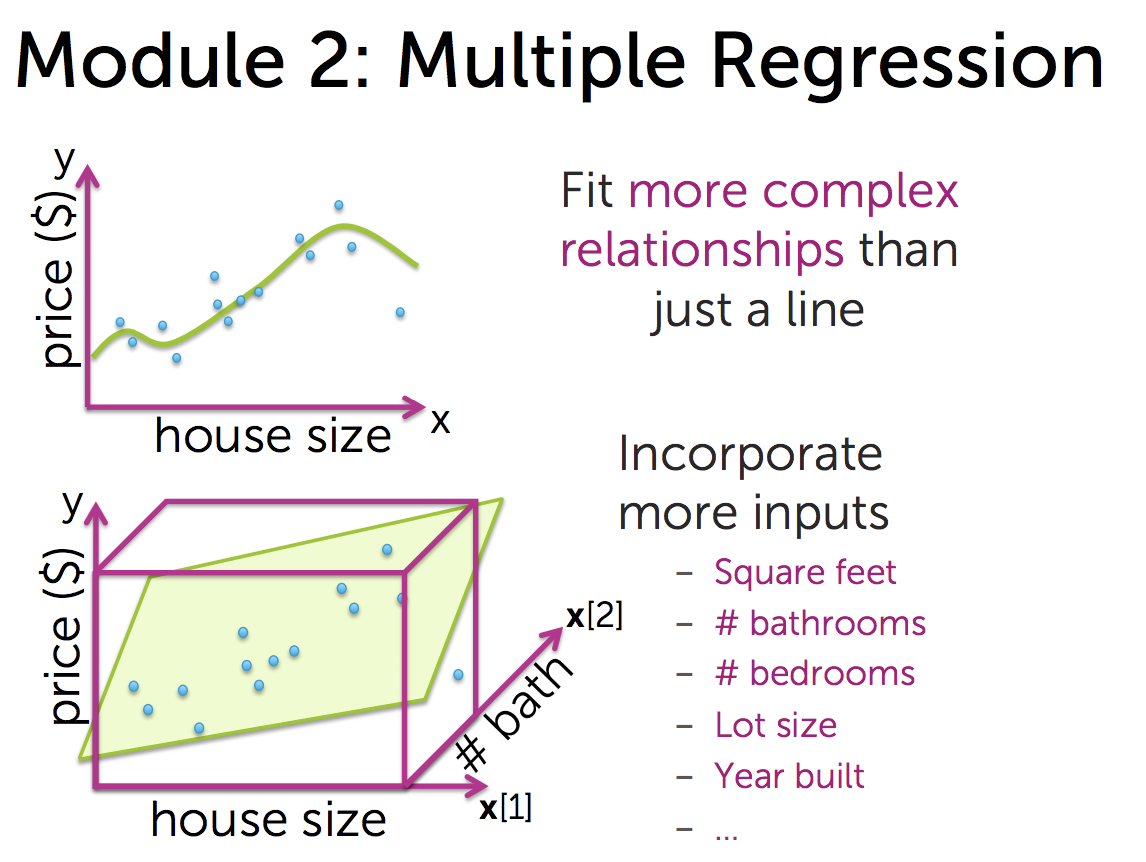
(an iterative algorithm to determine goodness-of-fit metric)

This method helps us to find the optimal **slope** and the **intercept** of the line for simple regression that defines goodness-of-fit metric.



1. Multiple Regression:

Fit more complex relationships than just a simple line. Incorporate more inputs (features)

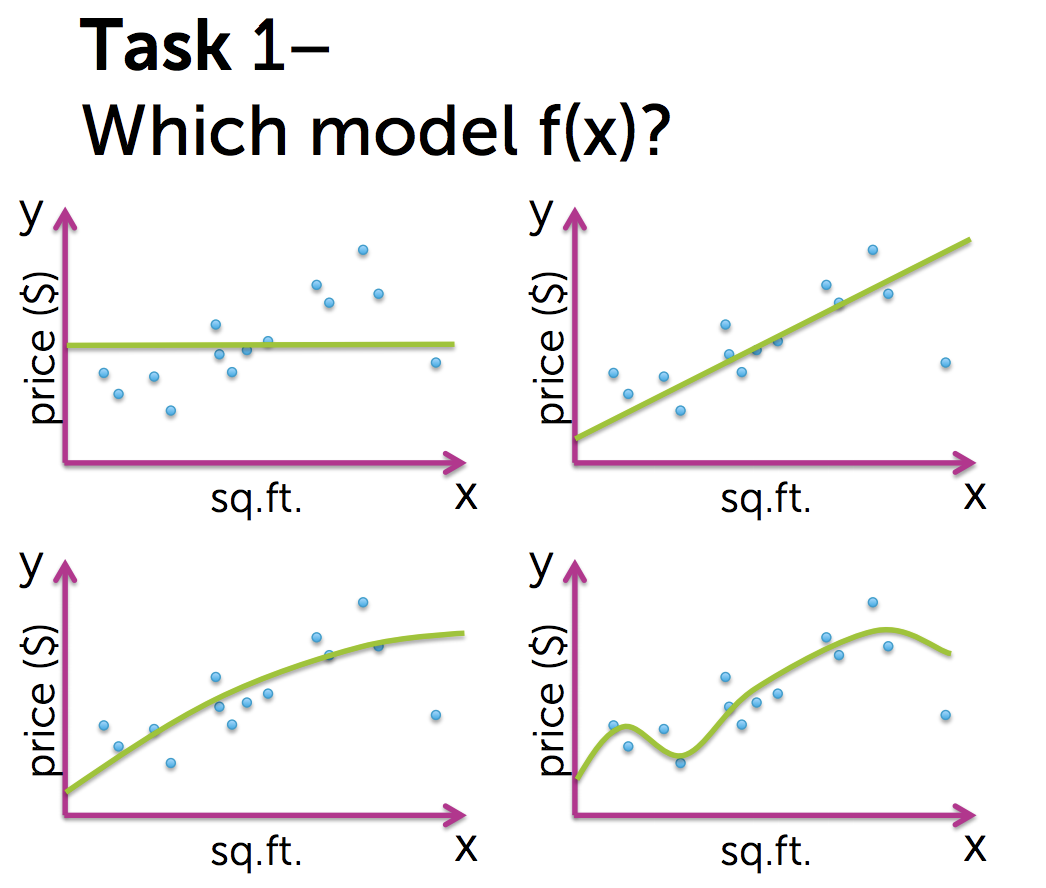


1. Assessing Performance:
2. Ridge Regression:
3. Feature Selection & Lasso Regression:
4. Nearest Neighbor & Kernel Regression:

# Linear Regression

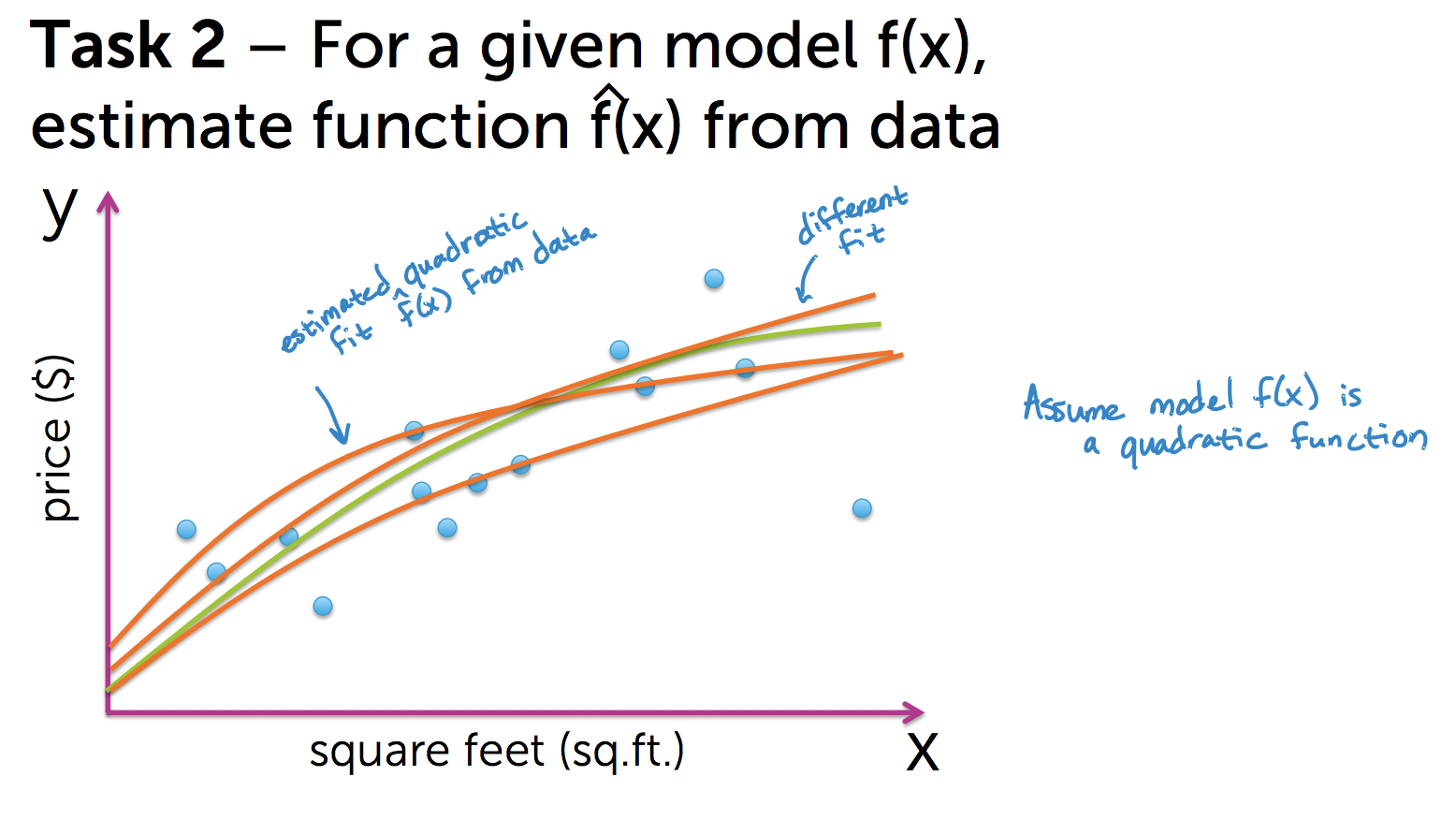
**Task 1:** Which model to select?

* Static line/ Constant model?
* Linier model?
* Quadratic model?
* Higher order polynomial model?

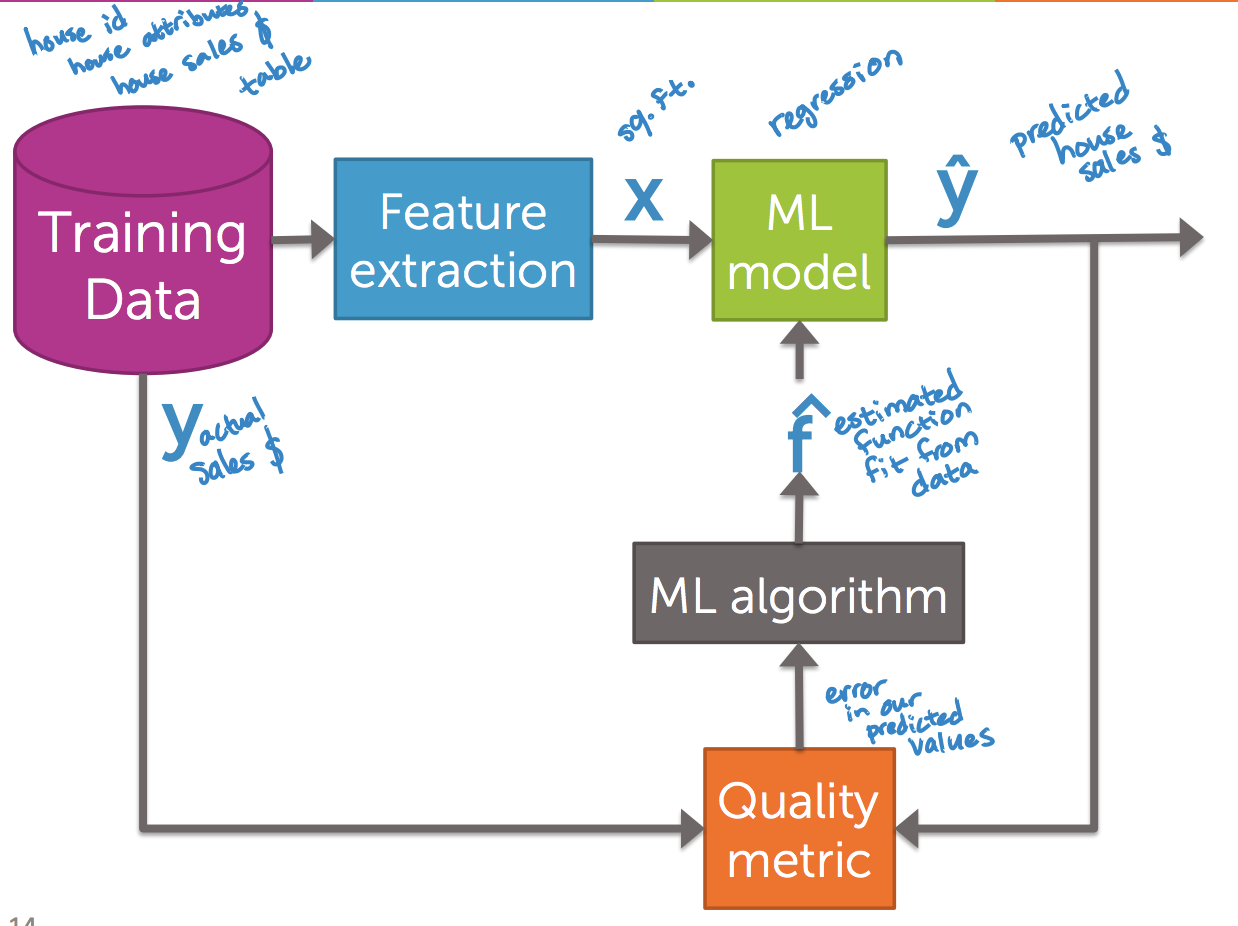


**Task 2:** For a given model , estimate function (x) from data.

Assume we select “quadratic fit”, and f(x) is a quadratic function. Then we will fit our data to different quadratic functions with different parameters and our task is to find the optimal parameters that provides a specific quadratic fit (x) to the data.



**ML Block Diagram (Regression):**



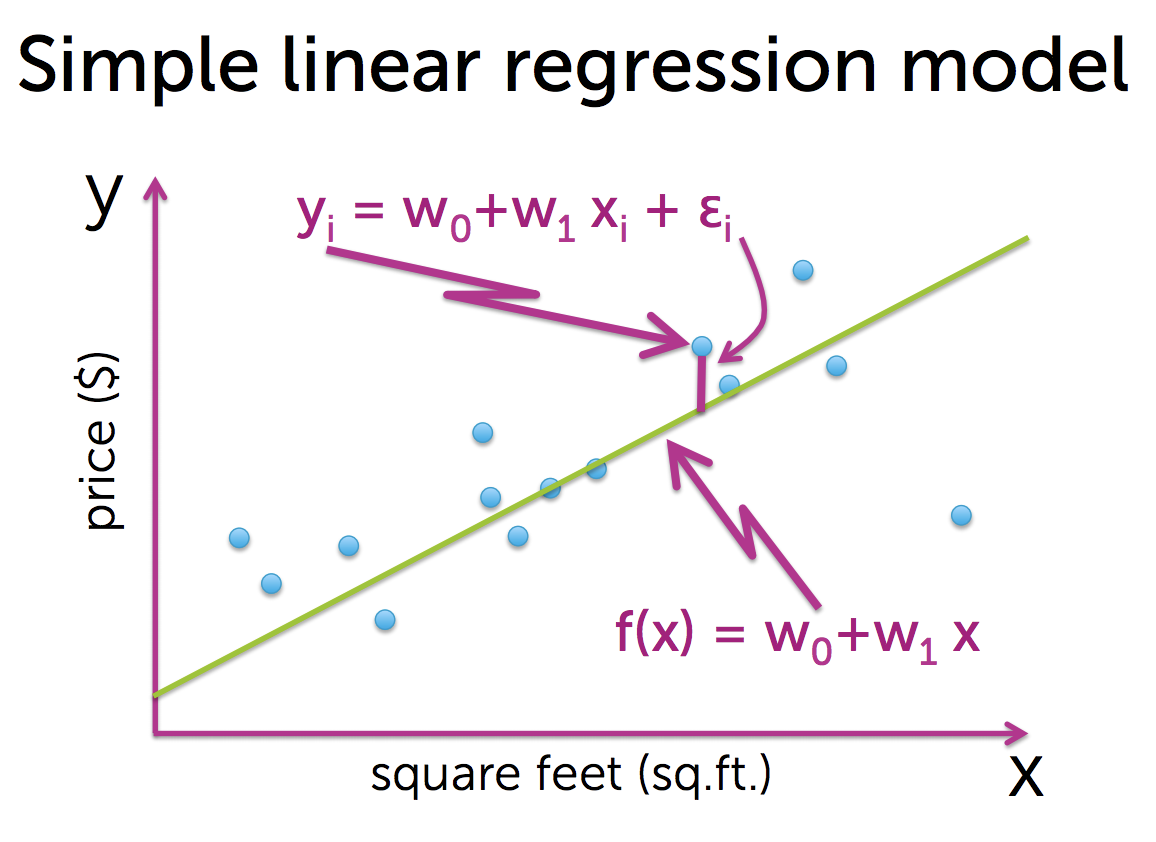
Training Data:

Feature Extraction:

ML Model:

ML Algorithm:

Quality Metric:

**Linear Regression Model:**

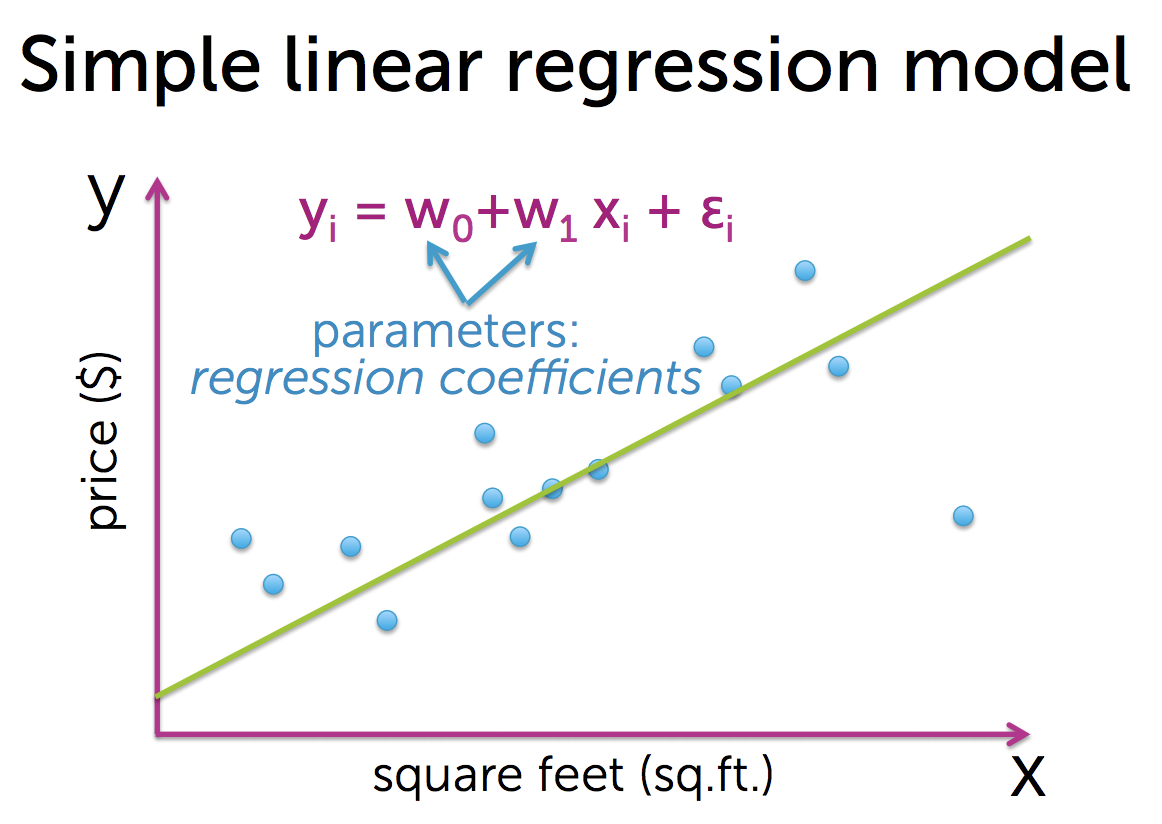
Function for line:

: Intercept

: Slope

: Single input (e.g. House size)

**Simple linear regression model:**



: Parameter (intercept)

: Parameter (slope)

: i th input input (e.g. i th house size)

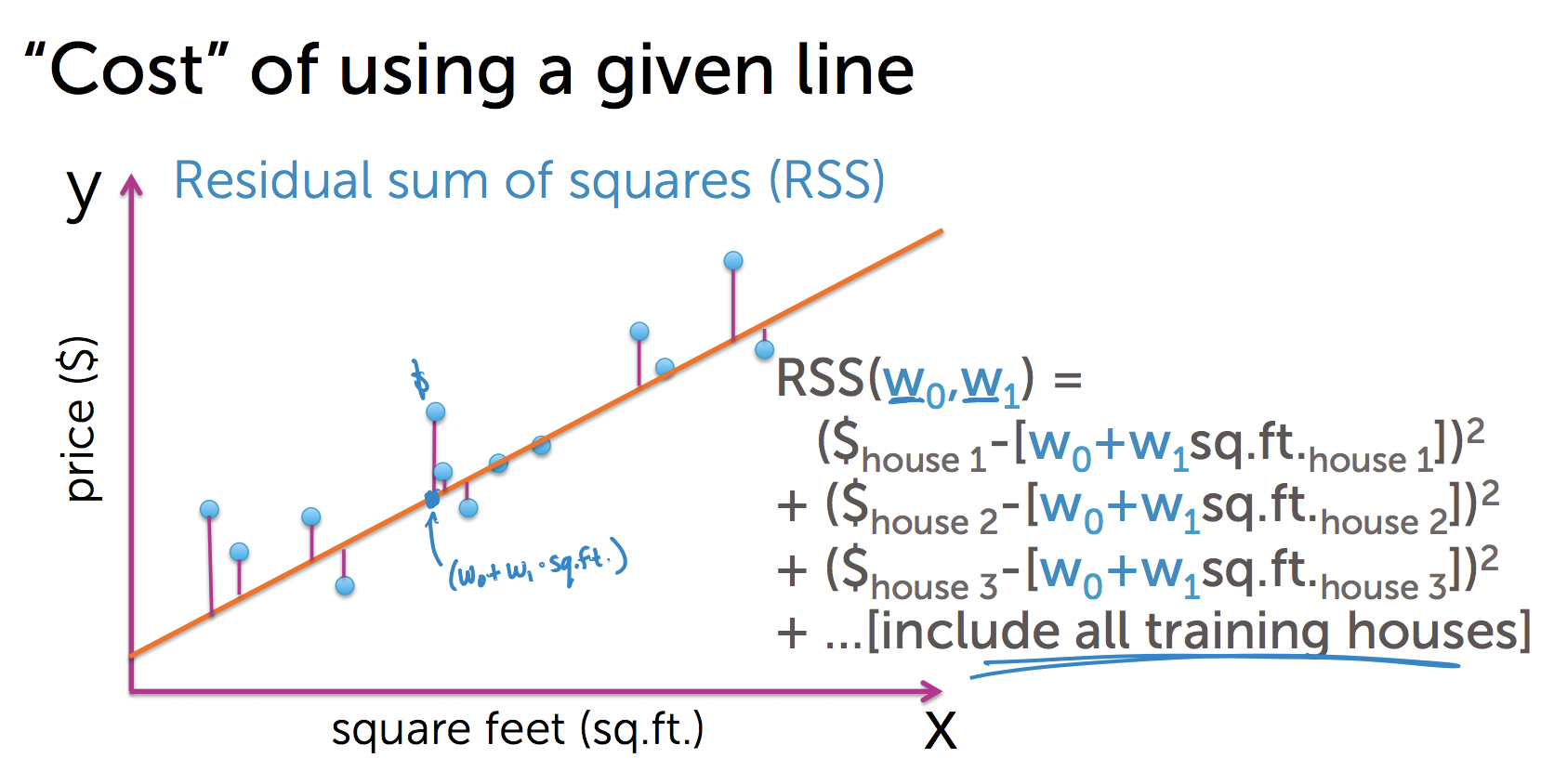
: error for the i th input

**and are called “regression coefficient”**

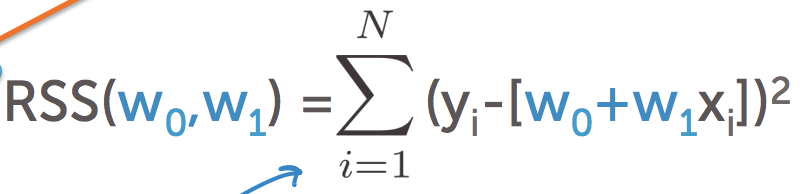
**[Quality Metric]**

Fitting a line to the data: (Finding the optimal line)

Cost of using a specific line: Accumulated RSS for each training data point.



Instead of writing RSS in this long equation, we could write it as:



We will calculate the accumulated “Residual sum of squares (RSS)” for each possible line that we can draw through our dataset, and then we’ll take the line that produces the minimum RSS. Then the parameters of that line will be our fitted parameters for our fitted function ( ).

(Note)

* Error () is a part of the model
* Residual is the difference between a prediction and the actual value in y axis.

**[ML Algorithm]**

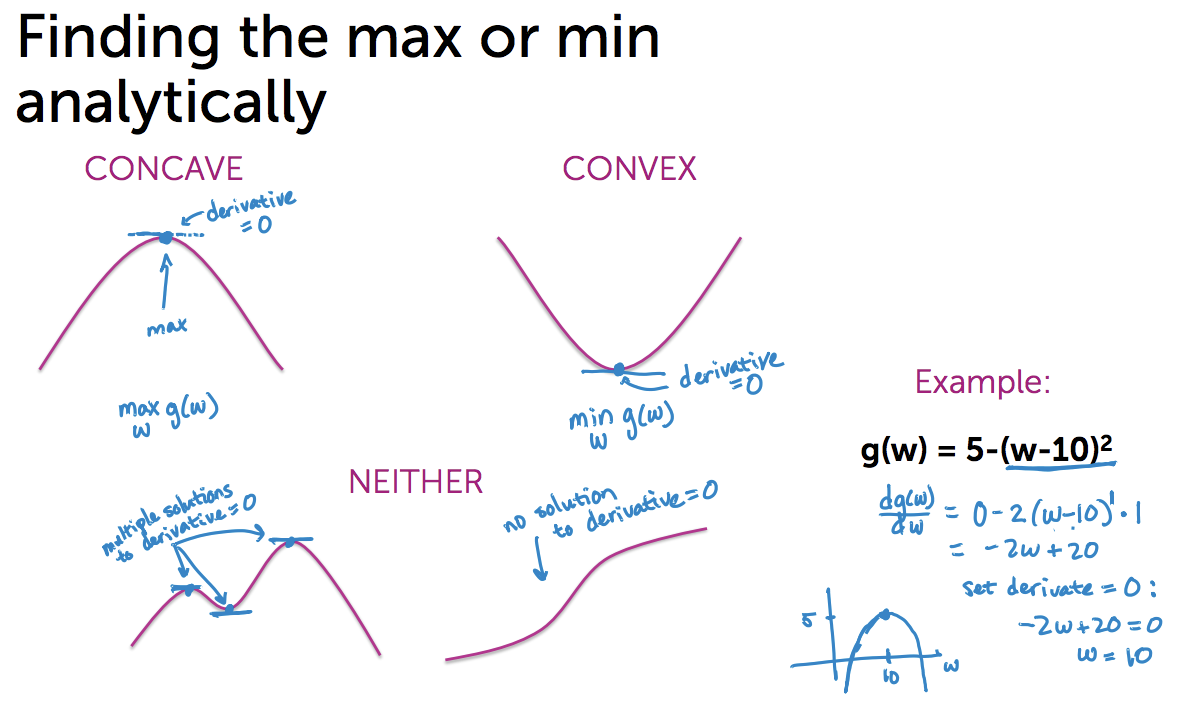
Good news is there is an algorithm (**Gradient** **descent**) that can go through all the possible lines and find the line that produces the minimum RSS for our dataset.

**Finding the maxima or minima (Optimization problem):** “convex” and “concave” function is used to find the max or min point of a curve. The ground the convex or concave function work is that “there's no rate of change of the function drawing the curve at its maximum or minimum point. Which means that the derivative of the function is Zero.

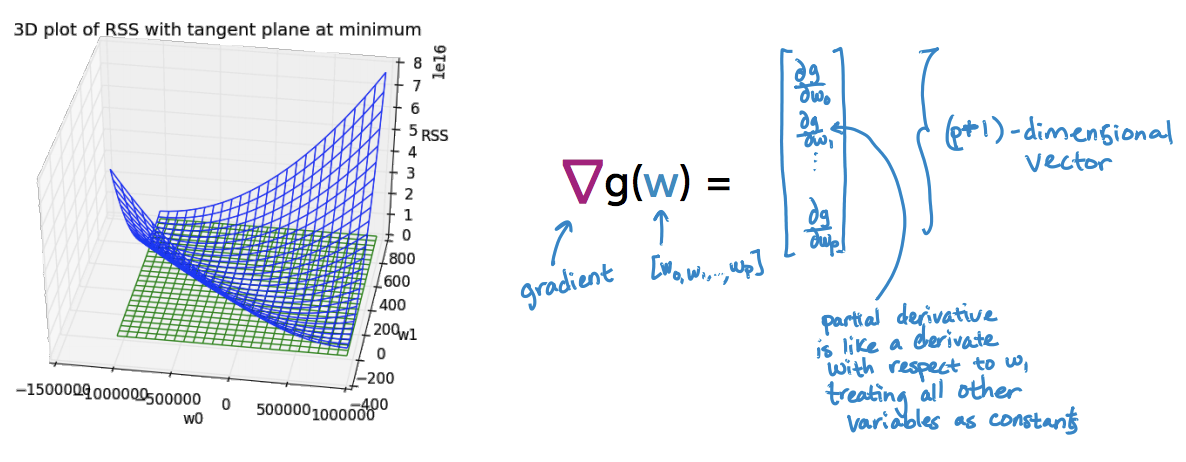
Example:

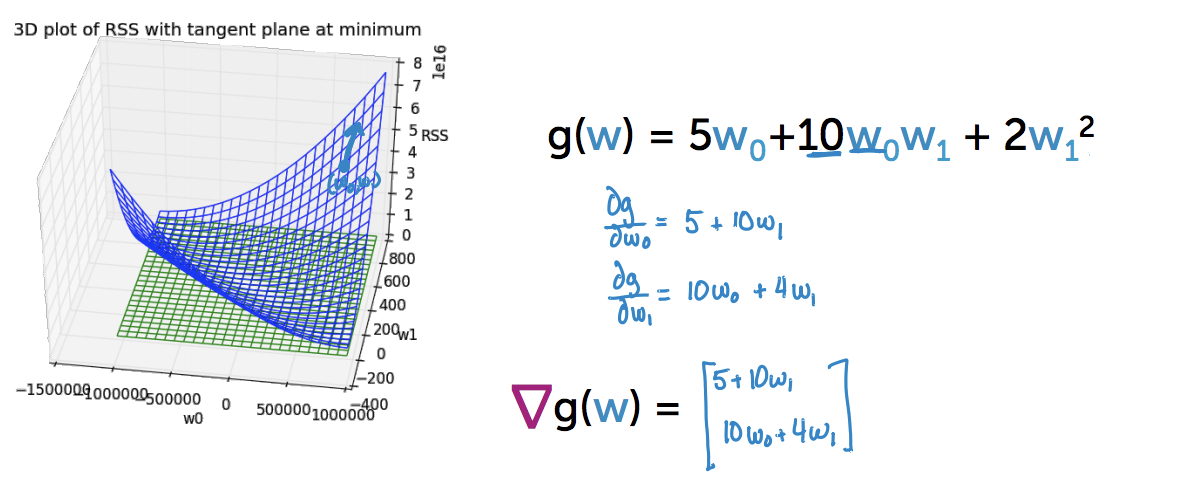
Set derivative equals to zero:

Which means that at 10 this function reaches the maximum.

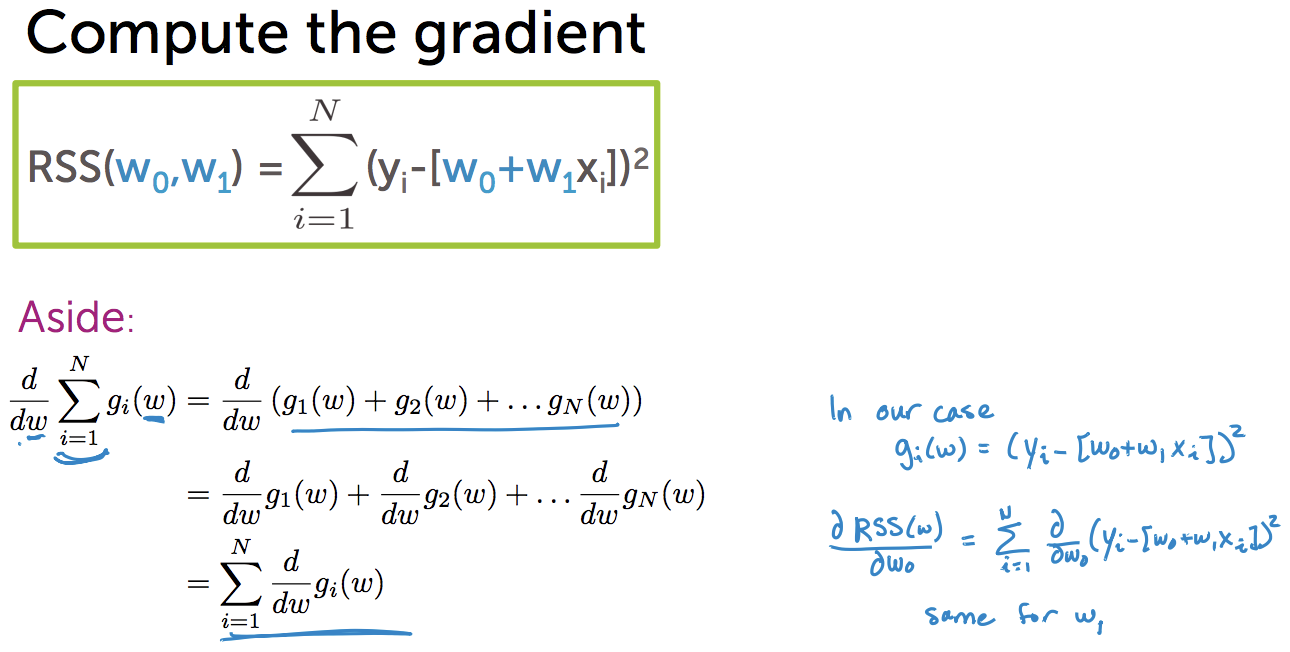


**Findig maximum or minimum for multiple dimensions (Gradients):** For multiple variables we don’t talk about derivatives anymore. We calculate “gradients”, which is actually a partial derivateves of each element of the array of variables.



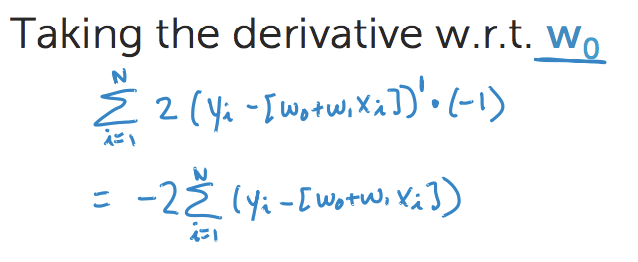


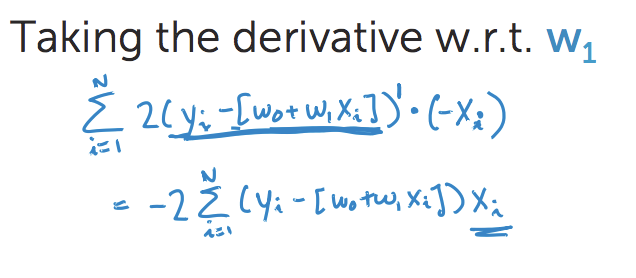
**Computing the gradient of RSS (Find the best line):**

****

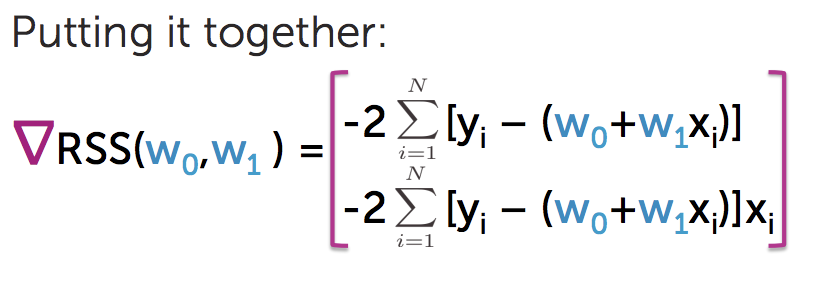
In our case we have two parameters ( and ). So in order to calculate the gradient descent, first we have to calculate partial derivative of these two parameters seperately. And then calculate the gradient descent.

**Step 1:** Compute partial derivatives of and :





**Step 2:** Calculating the gradient:



Approach 1: Set gradient = 0 and solve the equation

Approach 2: walking down slowly of residual sum of squares and try to get to the minimum.

(Note)

* For most ML problems we can’t solve gradient = 0.
* Even if solving gradient =0 is feasible, gradient descent can be more efficient.
* Gradient descent relies on chhoosing ***stepsize*** and ***convergence*** criteria.