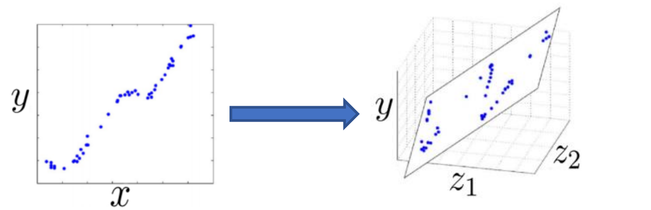
Linear Regression

Linear Regression: Pictorially

- Linear regression is like fitting a line or (hyper)plane to a set of points

$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \phi(x)$

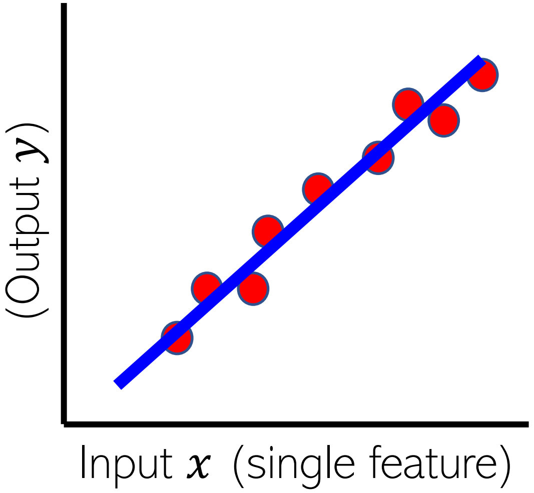


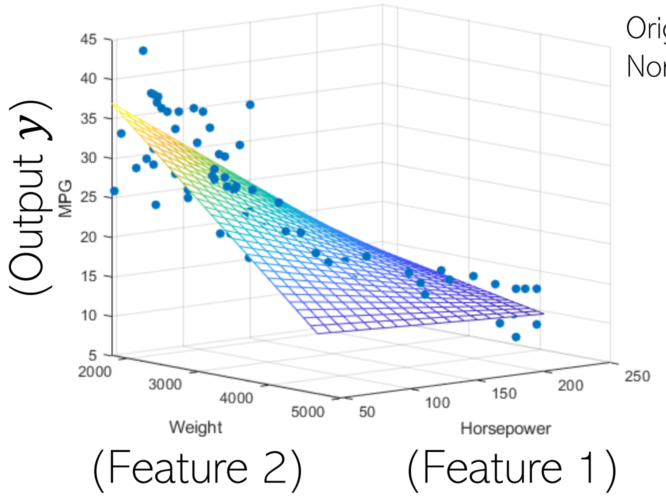
Original (single) feature
Nonlinear curve needed

Two features
Can fit a plane (linear)

What if a line/plane doesn't model the input-output relationship very well, e.g., if their relationship is better modeled by a nonlinear curve or curved surface?

Do linear models become useless in such cases?





No. We can even fit a curve using a linear model after suitably transforming the inputs

$$y \approx \boldsymbol{w}^\top \phi(\boldsymbol{x})$$

The transformation $\phi(.)$ can be predefined or learned (e.g., using kernel methods or a deep neural network based feature extractor). More on this later

- The line/plane must also predict outputs the unseen (test) inputs well

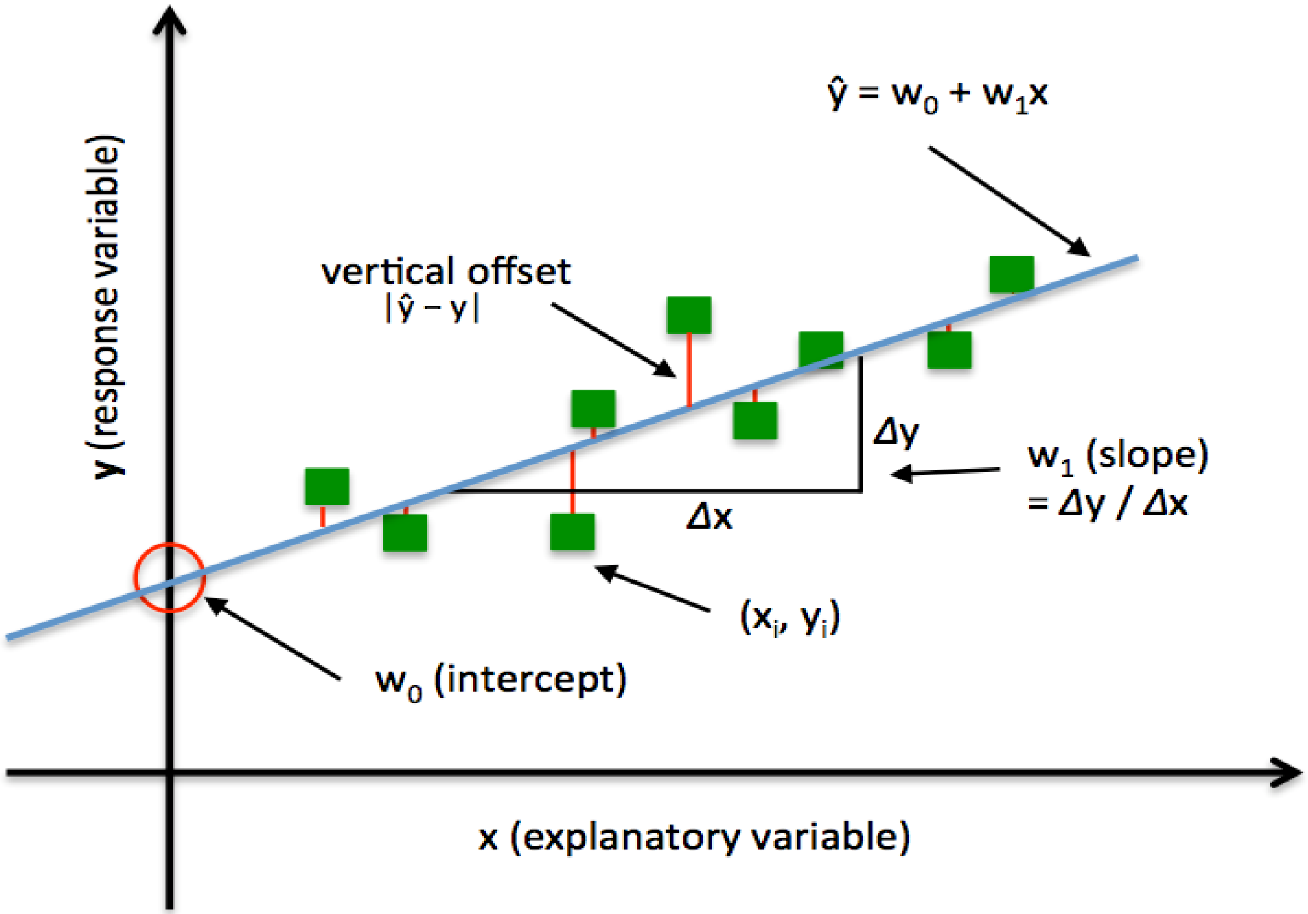
Simplest Possible Linear Regression Model

- This is the base model for <u>all</u> statistical machine learning
- x is a one feature data variable
- y is the value we are trying to predict
- The regression model is

$$y = w_0 + w_1 x + \varepsilon$$

Two parameters to estimate – the slope of the line w_1 and the y-intercept w_0

- ε is the unexplained, random, or error component.

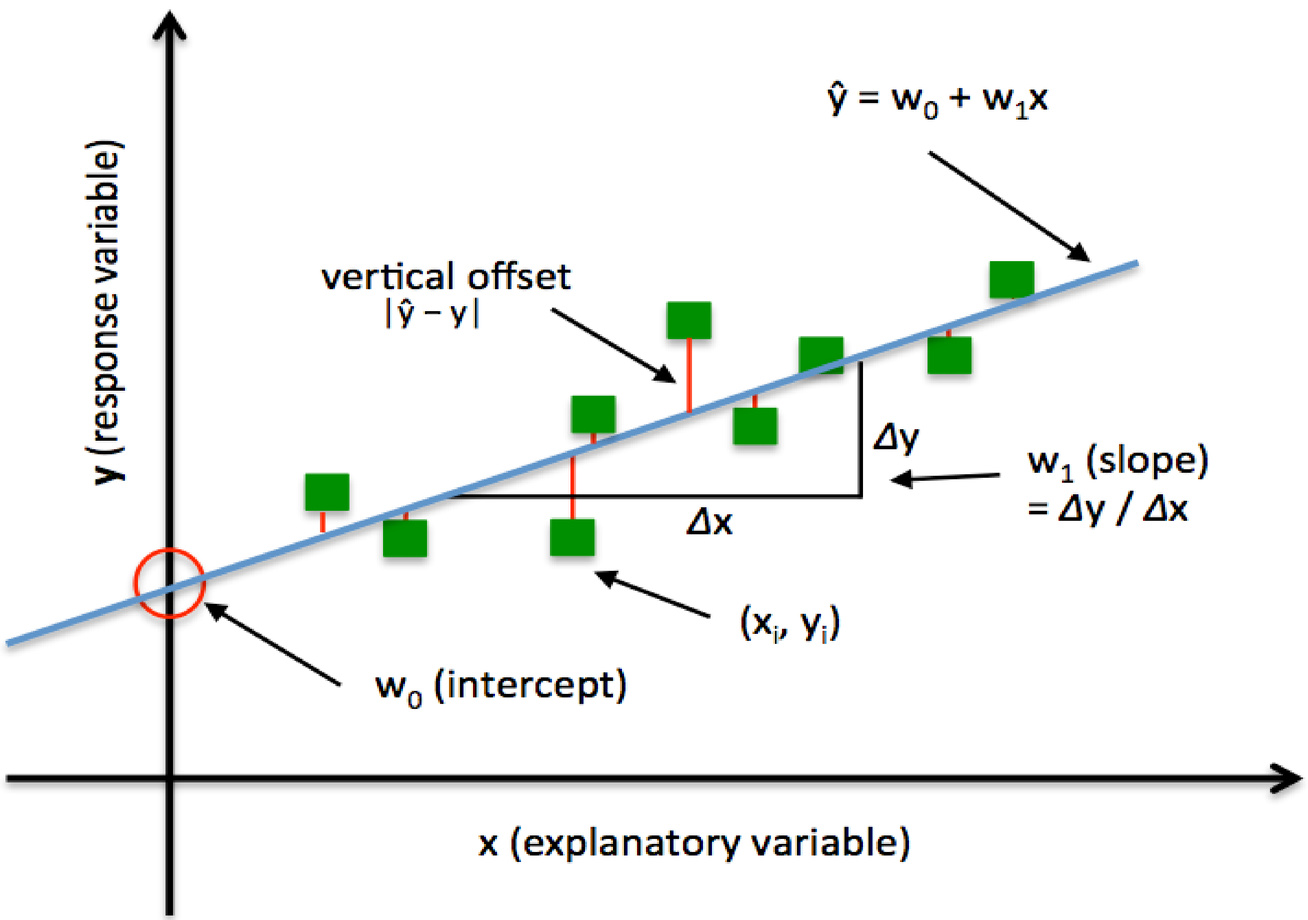


Solving the regression problem

- We basically want to find {w0, w1} that minimize deviations from the predictor line

$$\arg\min_{w_0,w_1} \sum_{i}^{n} (y_i - w_0 - w_1 x_i)^2$$

- How do we do it?
  - Iterate over all possible w values along the two dimensions?
  - Same, but smarter? [next class]
  - No, we can do this in *closed form* with just plain calculus
- Very few optimization problems in ML have closed form solutions
  - The ones that do are interesting for that reason



Parameter estimation via calculus

- We just need to set the partial derivatives to zero (full derivation)

$$\frac{\partial \epsilon^2}{\partial w_0} = \sum_i^n -2(y_i - w_0 - w_1 x_i) = 0$$

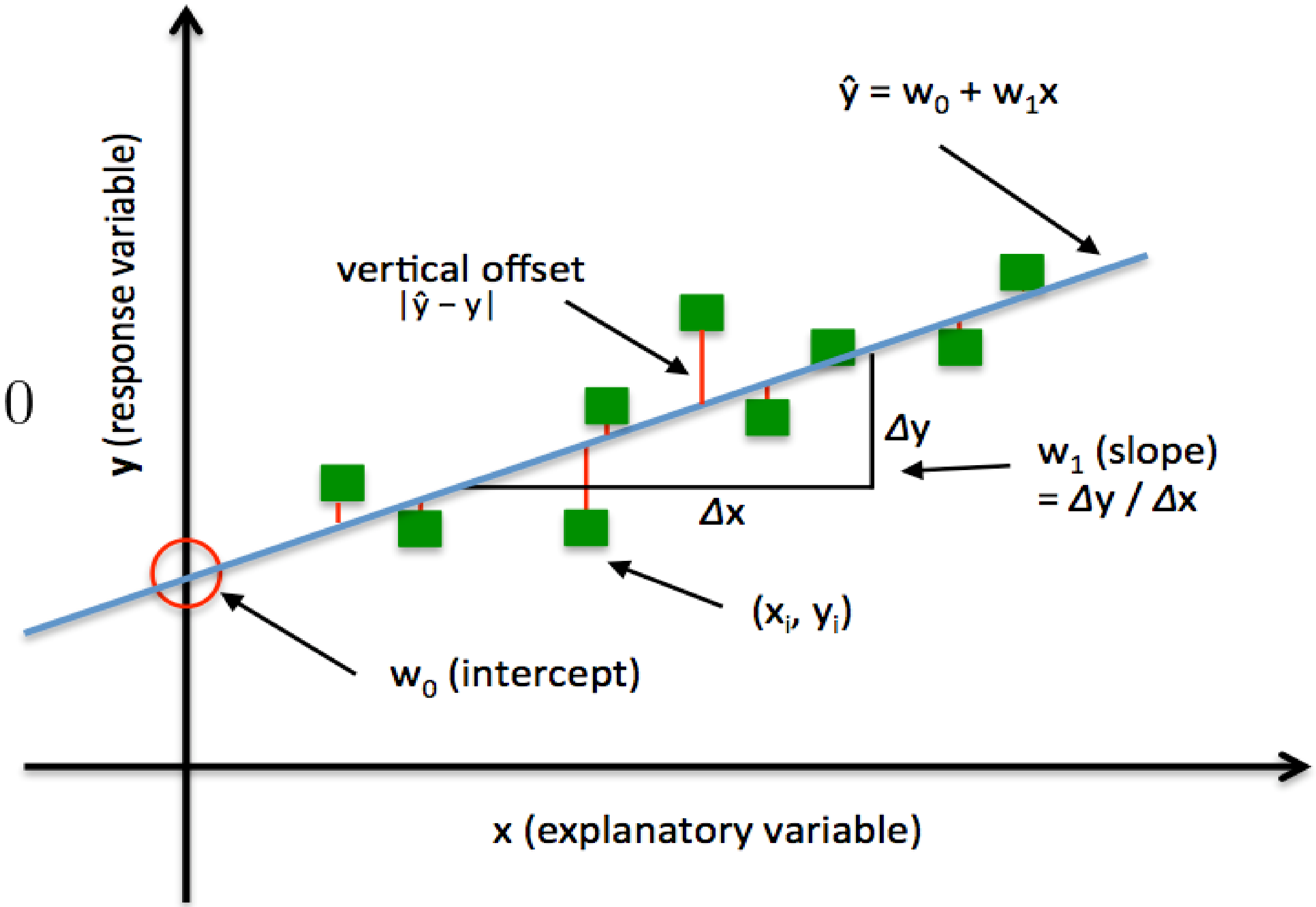
$$\frac{\partial \epsilon^2}{\partial w_1} = \sum_i^n -2x_i(y_i - w_0 - w_1 x_i) = 0$$

- Simplifying

$$w_0 = \bar{y} - w_1 \bar{x}$$

$$w_1 = \frac{n \sum_i^n x_i y_i - \sum_i^n x_i \sum_i^n y_i}{n \sum_i^n x_i x_i - \sum_i^n x_i \sum_i^n x_i}$$

$$= \frac{(\overline{xy}) - (\bar{x})(\bar{y})}{(\overline{x_i^2}) - (\bar{x})^2}$$



Problem

Xi(week)	Yi (Sales in Thousands)
1	1.2
2	1.8
3	2.6
4	3.2
5	3.8

Evaluation

x	y	x^2	$x * y$
1	1.2	1	1.2
2	1.8	4	3.6
3	2.6	9	7.8
4	3.2	16	12.8
5	3.8	25	19
Average=15/5=3	Average=12.6/5=2.52	Average=55/5=11	Average=44.4/5=8.88
Slope=$(8.88-3(2.52))/(11-3^2)=0.66$	Intercept=2.52-0.66*3= 0.54	Line: y=0.54-0.66*x	

Validation

- Standard Error: difference between actual and predicted value
- Mean Absolute Error(MAE) : $\frac{1}{n}\sum_{i=0}^{n-1}|y_i - \widehat{y_i}|$
- Mean squared Error (MSE): $\frac{1}{n}\sum_{i=0}^{n-1}(y_i - \widehat{y_i})^2$
- Root Mean Square Error(RMSE): Root of MSE
- Relative MSE: ratio of the prediction ability of y cap to the average of the trivial population
  - $\frac{\sum_{i=0}^{n-1}(y_i-\widehat{y_i})^2}{\sum_{i=0}^{n-1}(y_i-\overline{y_i})^2}$
- Coefficient of variation: RMSE/ $\bar{y}$