

MIDS W207

# Applied Machine Learning

Summer 2022

Week 2

# Regression Analysis

Regression analysis is a set of statistical methods used for the estimation of relationships between a dependent variable and one or more independent variables.

It can be utilized to assess the strength of the relationship between variables and for modeling the future relationship between them.

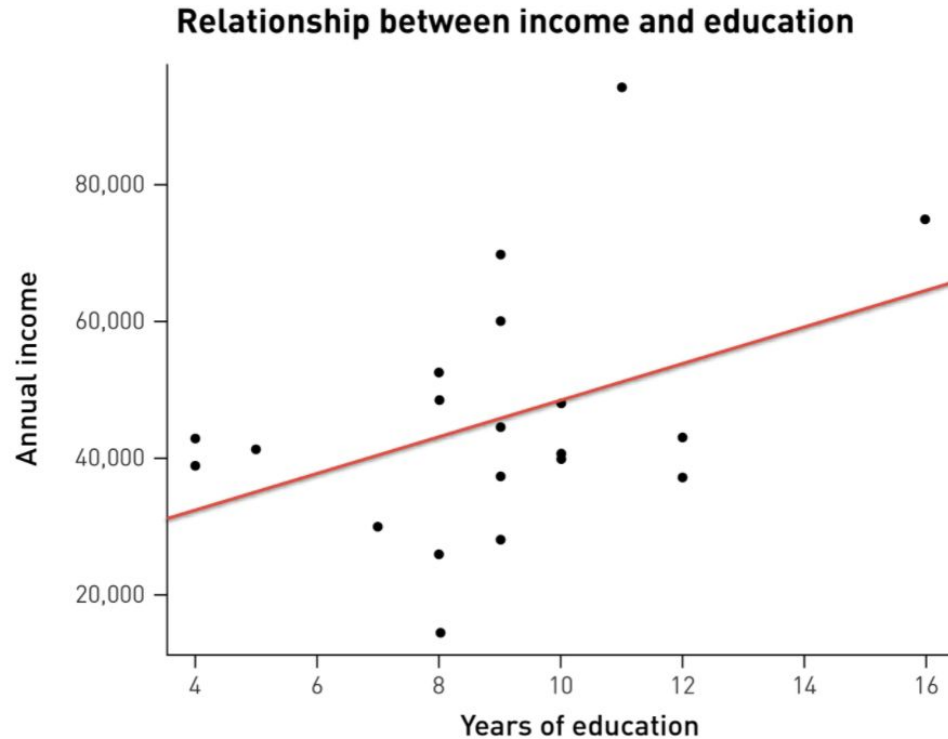
## Applications

1. Forecasting
2. Capital Asset Pricing Model (CAPM)
3. Comparing with competition
4. Identifying problems
5. Reliable Source

The diagram shows the linear regression equation  $Y_i = \beta_0 + \beta_1 X_i$  enclosed in a dashed red box. Arrows point from descriptive labels to each term in the equation:  $Y_i$  is labeled 'Dependent Variable' with an upward arrow;  $\beta_0$  is labeled 'Constant/Intercept' with a downward arrow;  $\beta_1$  is labeled 'Slope/Coefficient' with an upward arrow; and  $X_i$  is labeled 'Independent Variable' with a downward arrow.

$$Y_i = \beta_0 + \beta_1 X_i$$

# Regression Analysis: Example



# Regression Analysis: Notations

## Subscript Notation

$$y_i = \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = x_i^T \beta + \varepsilon_i$$

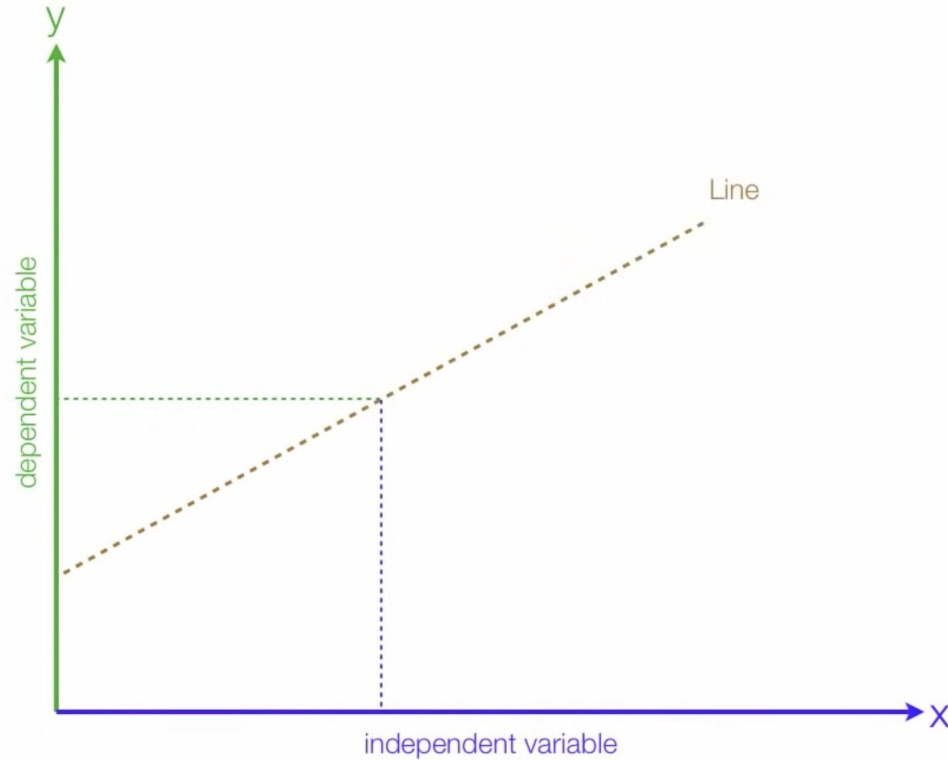
$$i = 1, \dots, n$$

## Matrix Notation

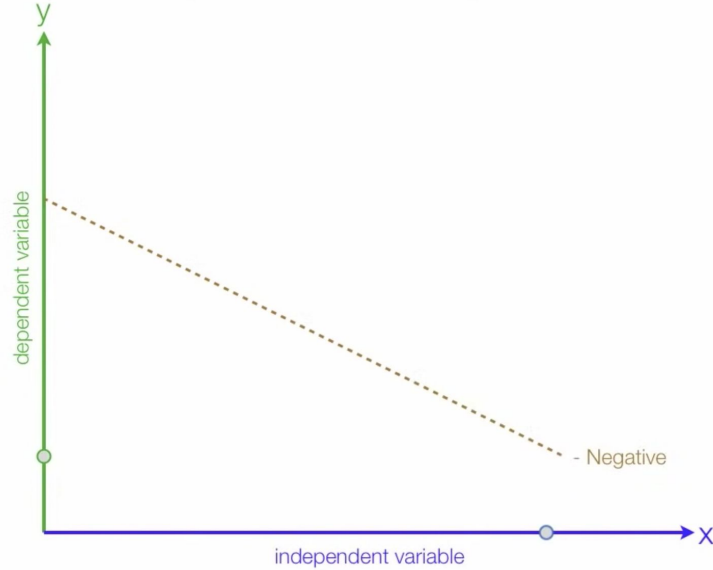
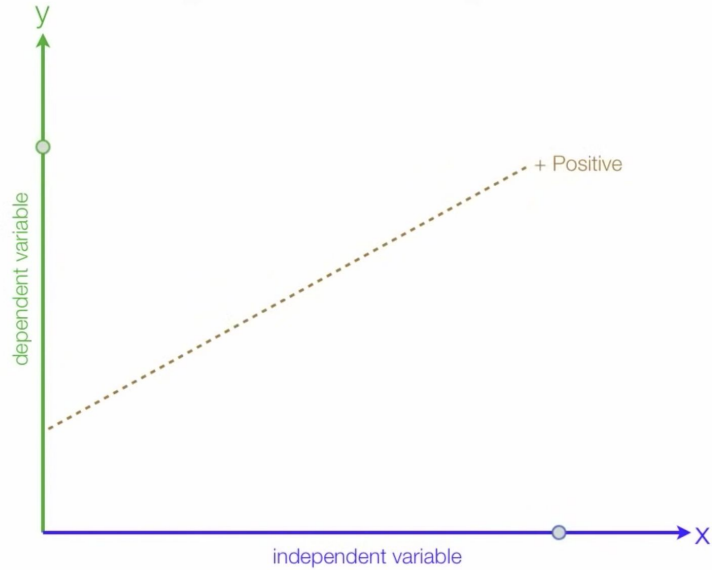
$$y = X\beta + \varepsilon$$

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad X = \begin{pmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_n^T \end{pmatrix} = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ x_{21} & \cdots & x_{2p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{pmatrix}, \quad \beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix}, \quad \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

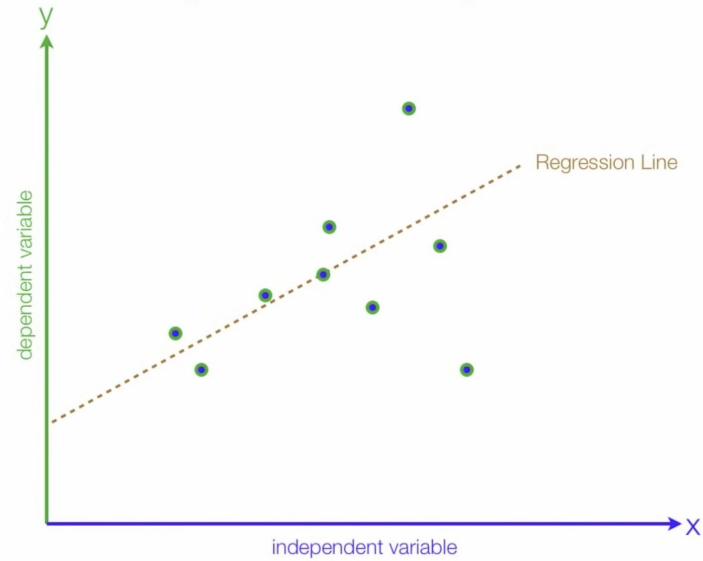
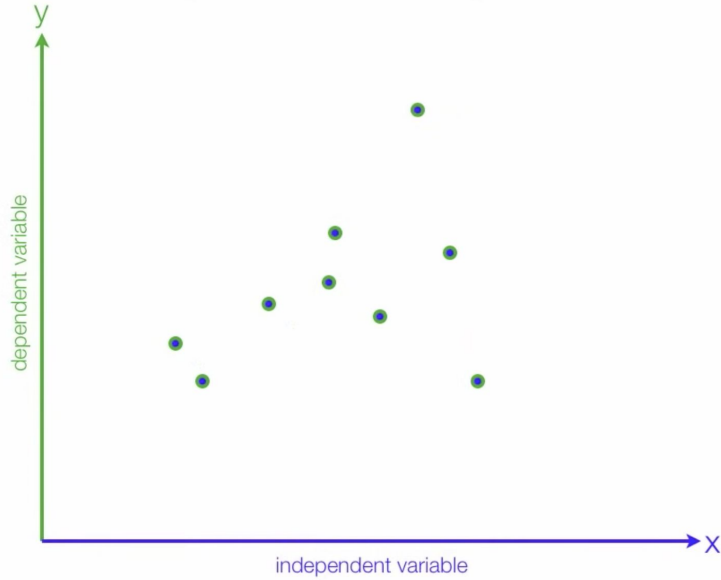
# Regression Example



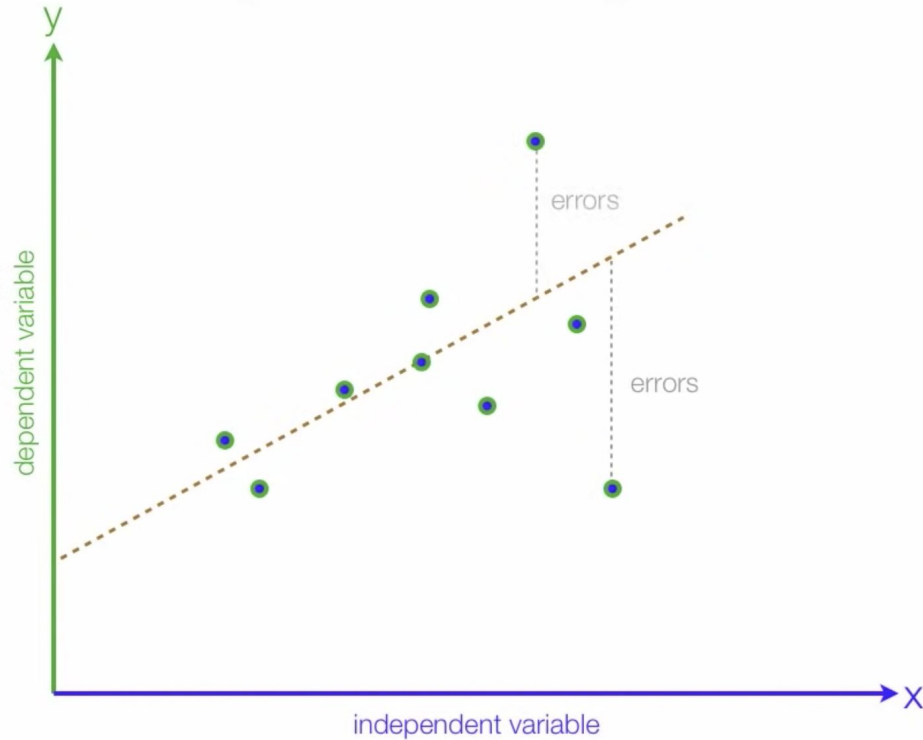
# Regression Example



# Regression Example

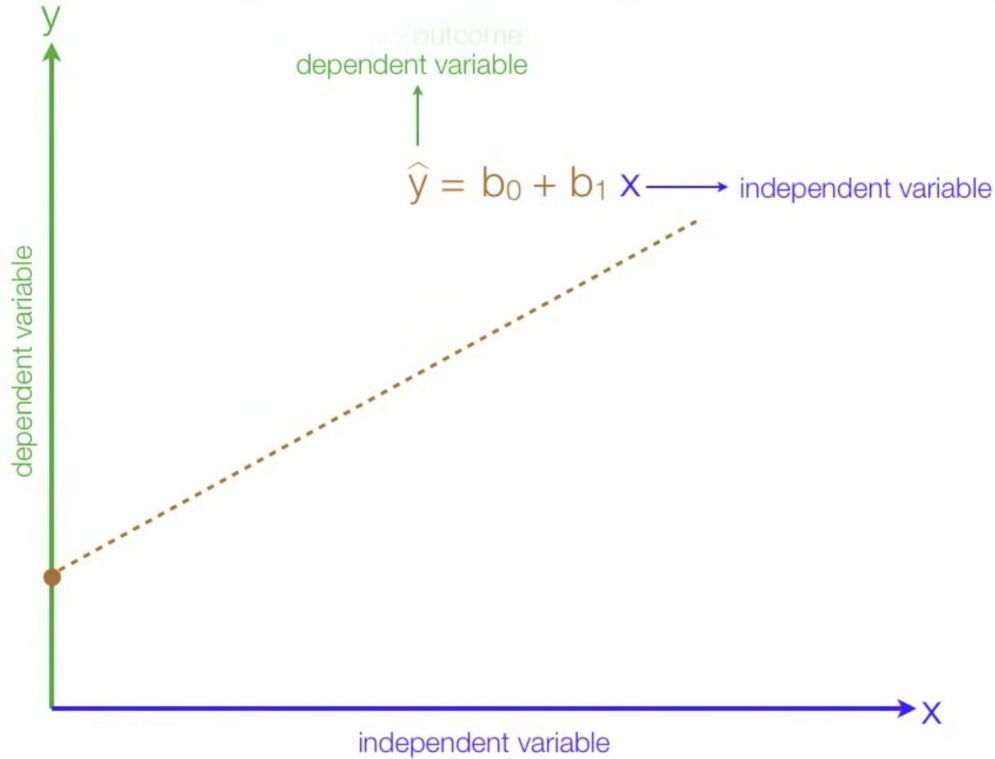


# Regression Example

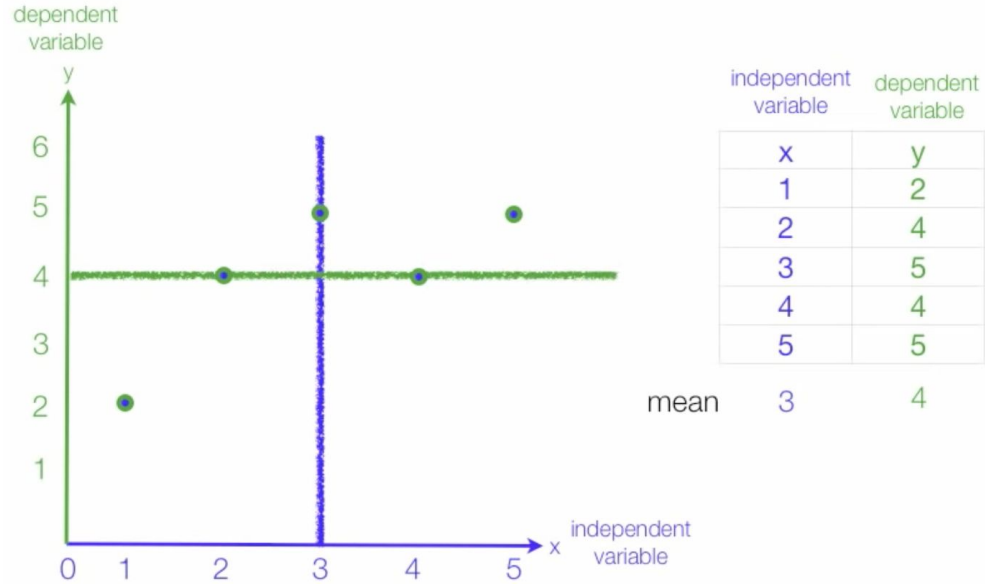




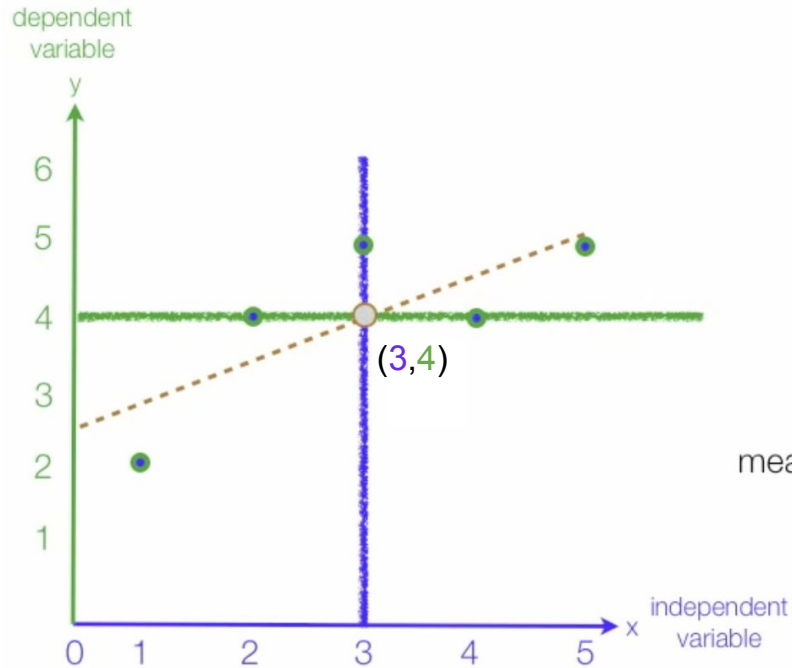
# Regression Example



# Regression Example



# Regression Example



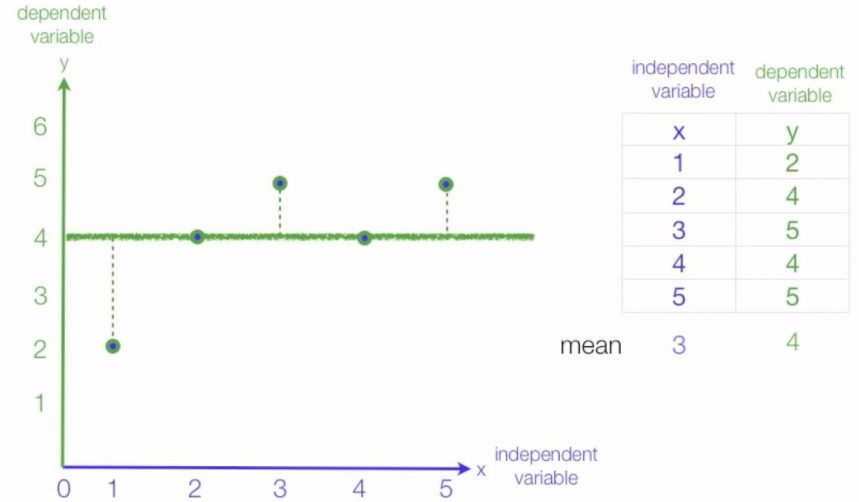
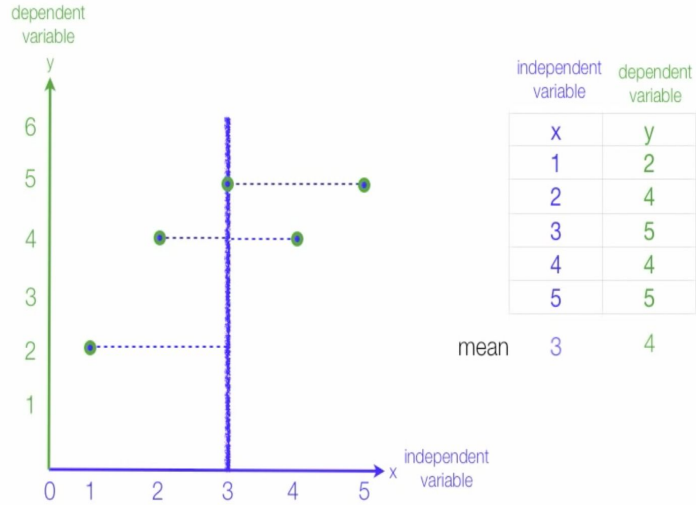
independent variable	dependent variable
x	y
1	2
2	4
3	5
4	4
5	5

mean

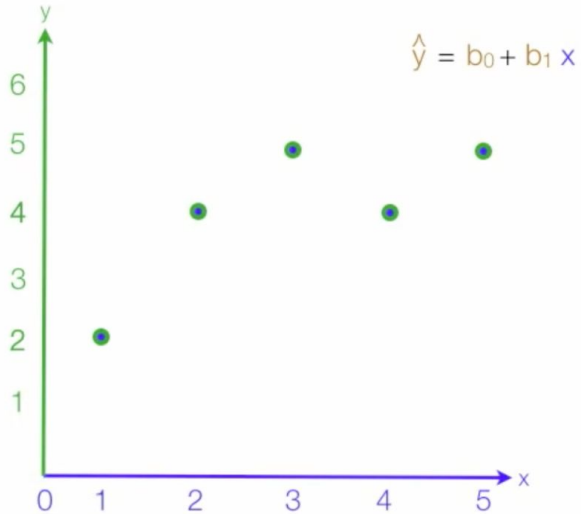
3

4

# Regression Example



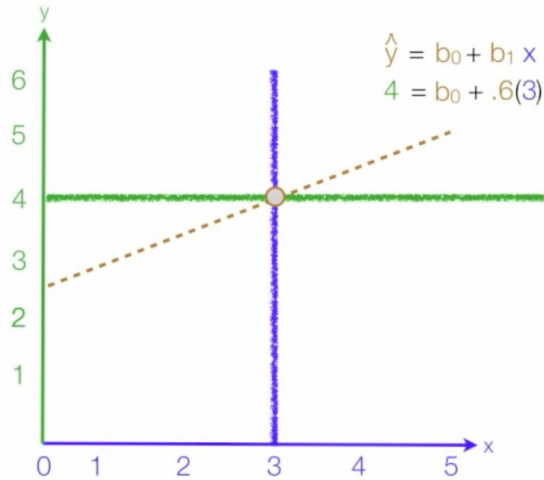
# Regression Example



x	y	$x - \bar{x}$	$y - \bar{y}$	$(x - \bar{x})^2$	$(x - \bar{x})(y - \bar{y})$
1	2	-2	-2	4	4
2	4	-1	0	1	0
3	5	0	1	0	0
4	4	1	0	1	0
5	5	2	1	4	2

mean    3    4

# Regression Example



$$b_0 = 2.2$$

$$b_1 = .6$$

$$\hat{y} = 2.2 + .6x$$

x	y	$x - \bar{x}$	$y - \bar{y}$	$(x - \bar{x})^2$	$(x - \bar{x})(y - \bar{y})$
1	2	-2	-2	4	4
2	4	-1	0	1	0
3	5	0	1	0	0
4	4	1	0	1	0
5	5	2	1	4	2
mean		3	4	10	6

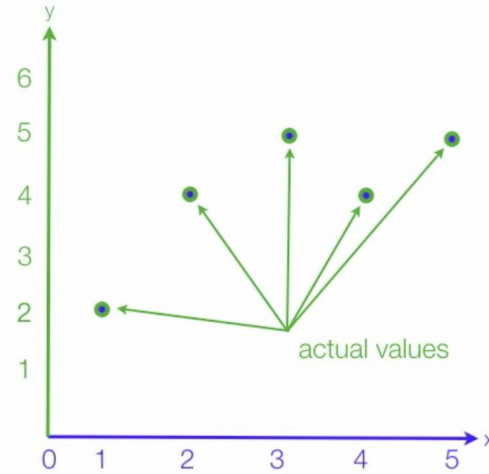
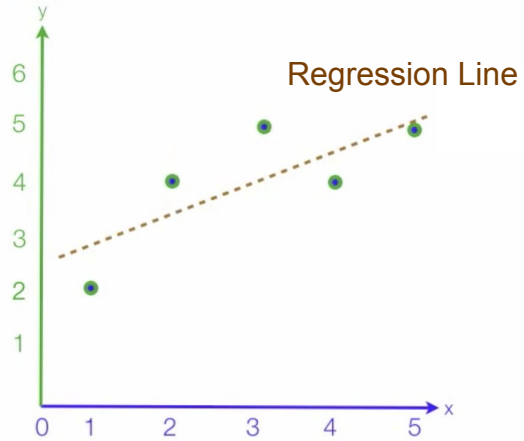
$$4 = b_0 + .6(3)$$

$$4 = b_0 + 1.8$$

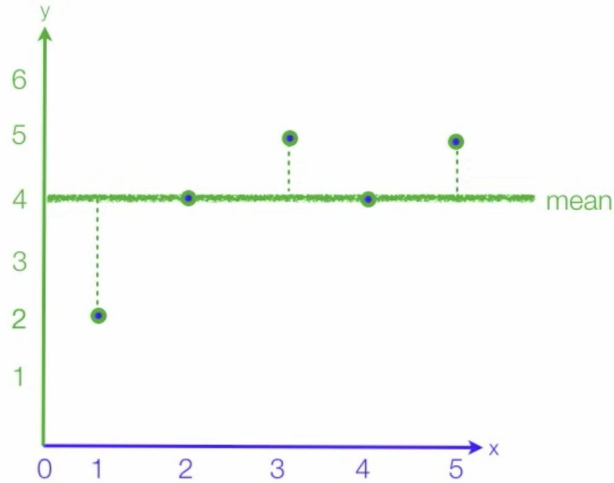
$$\begin{array}{r} 4 \\ -1.8 \\ \hline 2.2 = b_0 \end{array}$$

$$b_1 = \frac{6}{10} = .6 = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sum (x - \bar{x})^2}$$

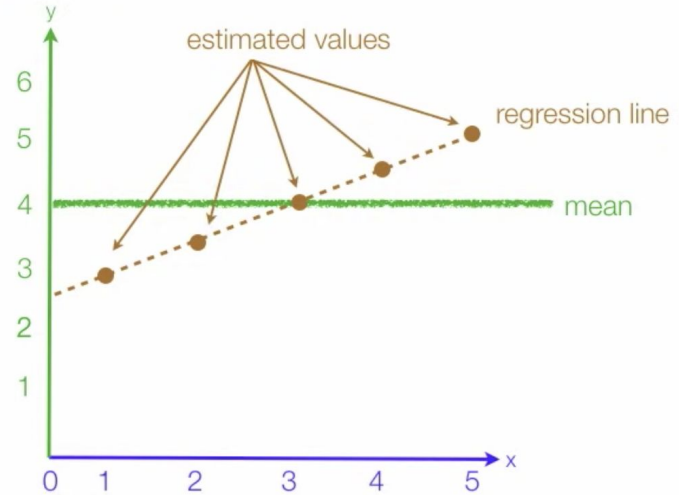
# Regression Example: R-Squared



# Regression Example: R-Squared



distance  
actual - mean

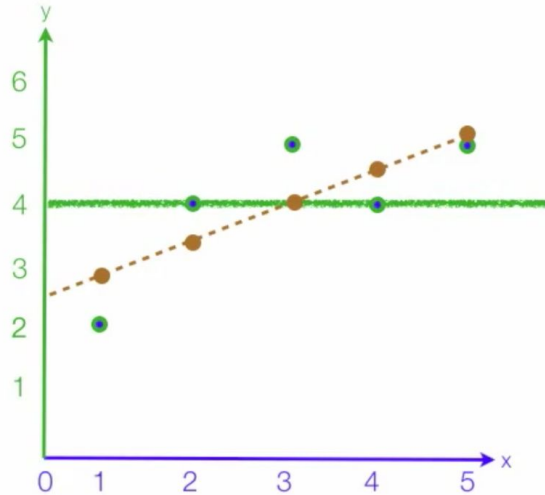


distance  
estimated - mean

compare



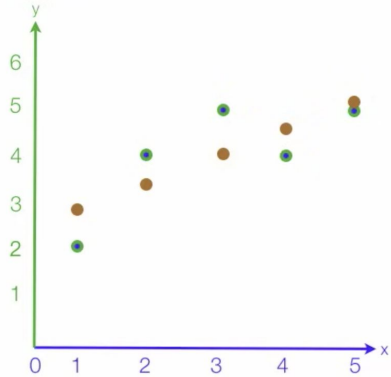
# Regression Example: R-Squared



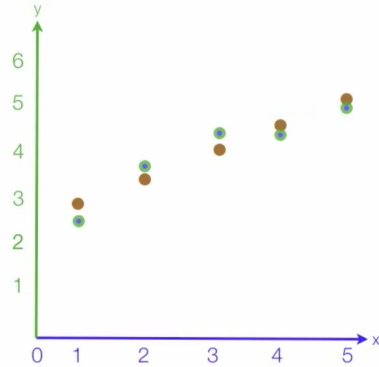
x	y	$y - \bar{y}$	$(y - \bar{y})^2$	$\hat{y}$	$\hat{y} - \bar{y}$	$(\hat{y} - \bar{y})^2$
1	2	-2	4	2.8	-1.2	1.44
2	4	0	0	3.4	-.6	.36
3	5	1	1	4	0	0
4	4	0	0	4.6	.6	.36
5	5	1	1	5.2	1.2	1.44
mean		4	6			3.6

$$R^2 = \frac{3.6}{6} = .6 = \frac{\sum (\hat{y} - \bar{y})^2}{\sum (y - \bar{y})^2}$$

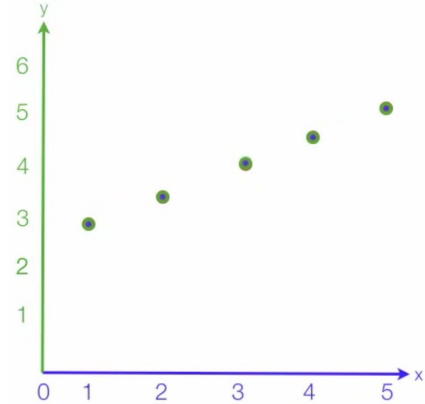
# Regression Example: R-Squared



$R^2 = .6$



$R^2 = .90$



$R^2 = 1$


# Gradient Descent

Gradient Descent is an optimization algorithm for finding a local minimum of a differentiable function.

Gradient descent is simply used in machine learning to find the values of a function's parameters (coefficients) that minimize a cost function as far as possible.

It's based on a convex function and tweaks its parameters iteratively to minimize a given function to its local minimum.

# Gradient Descent

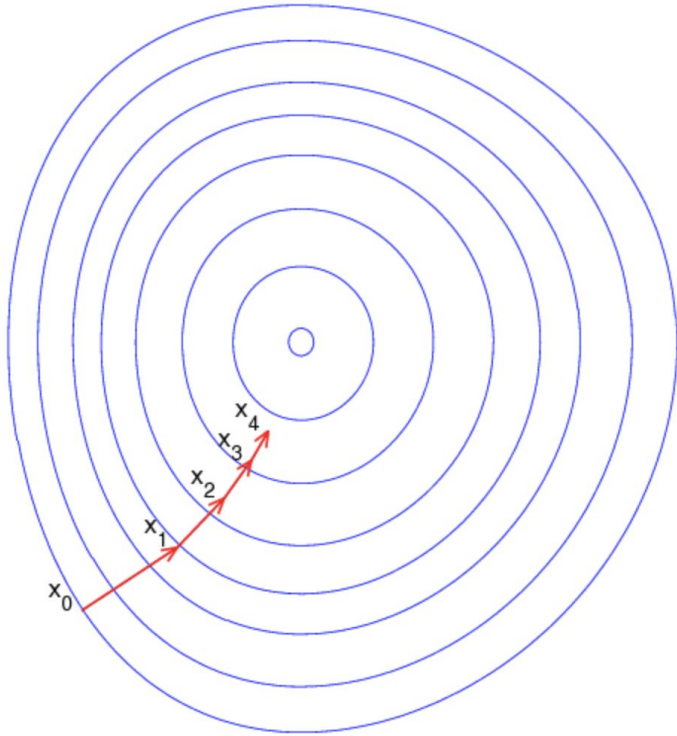


**"A gradient measures how much the output of a function changes if you change the inputs a little bit." —Lex Fridman (MIT)**

A gradient is a derivative of a function that has more than one input variable.

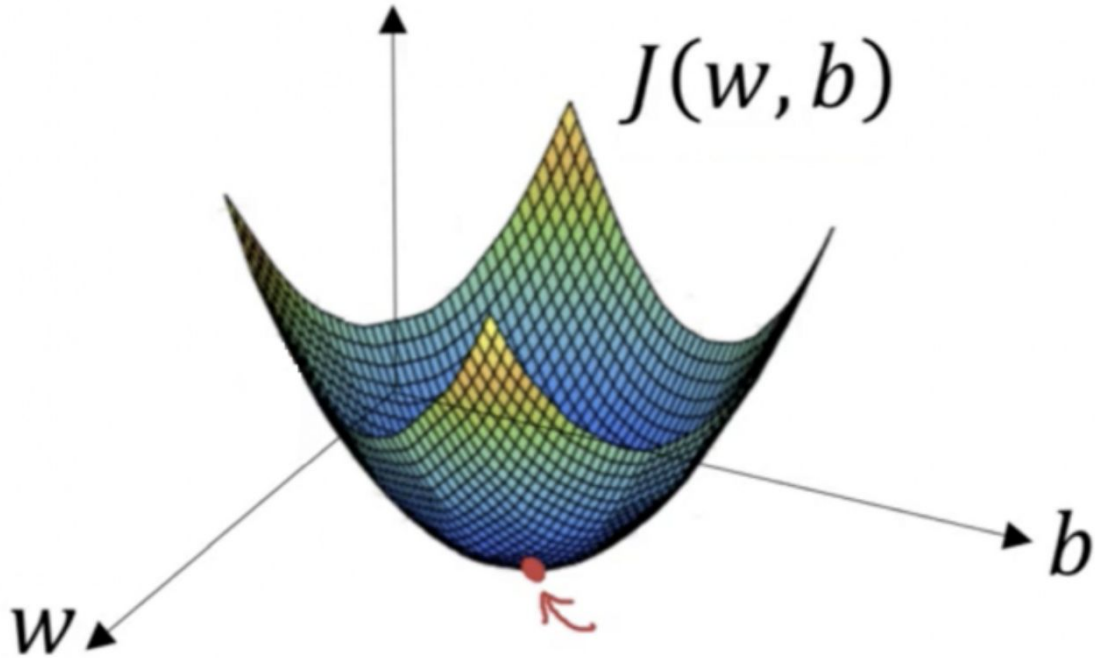
Known as the slope of a function in mathematical terms, the gradient simply measures the change in all weights with regard to the change in error.

# Gradient Descent

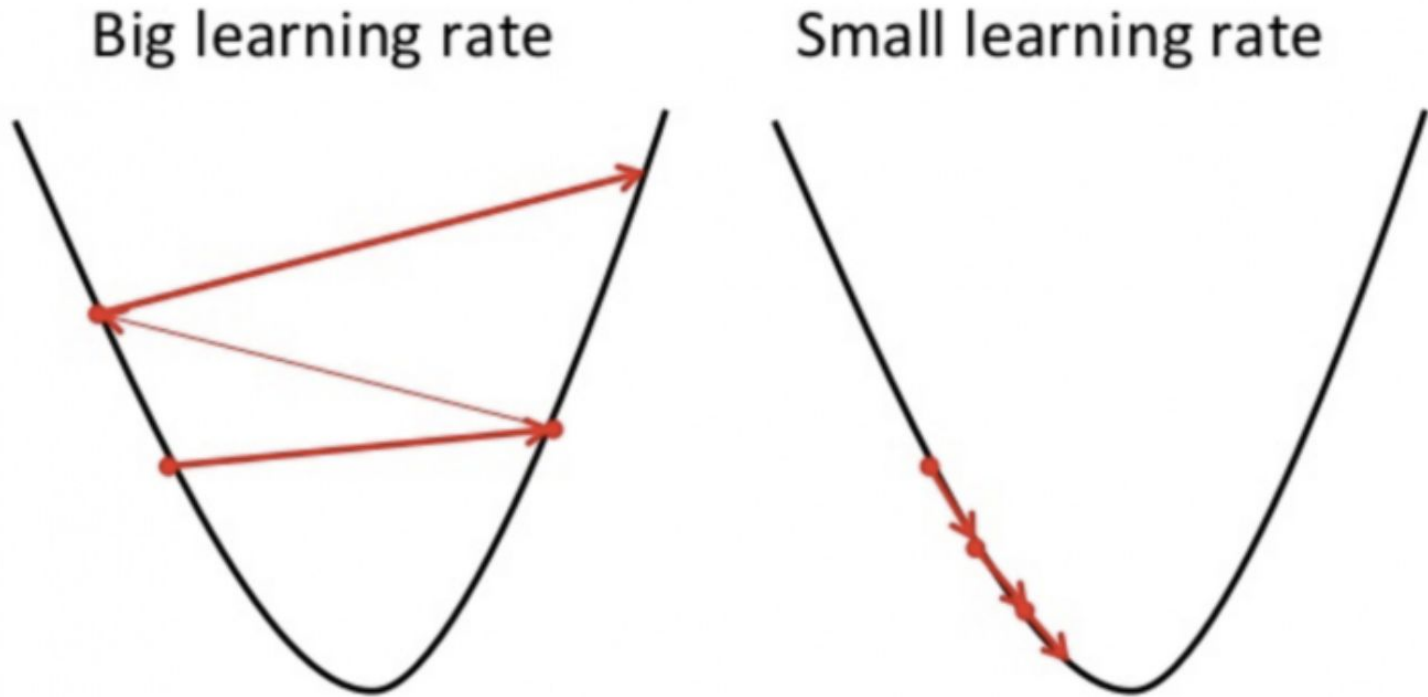


$$\mathbf{b} = \mathbf{a} - \gamma \nabla f(\mathbf{a})$$

# Gradient Descent: Analysis



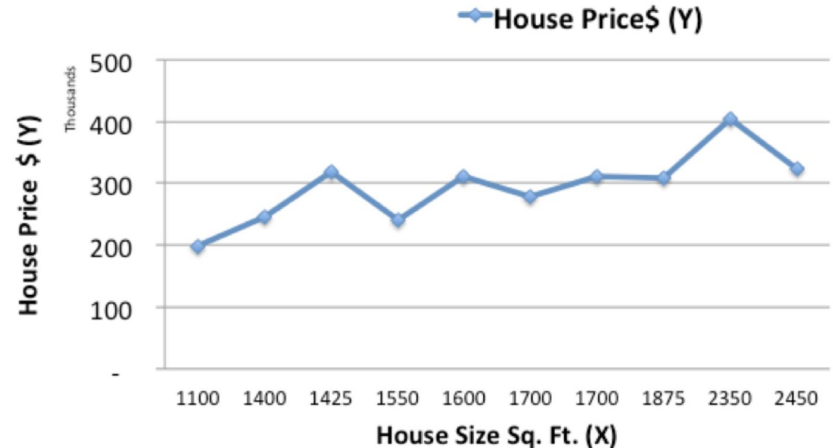
# Gradient Descent: Learning Rate



# Gradient Descent: Example

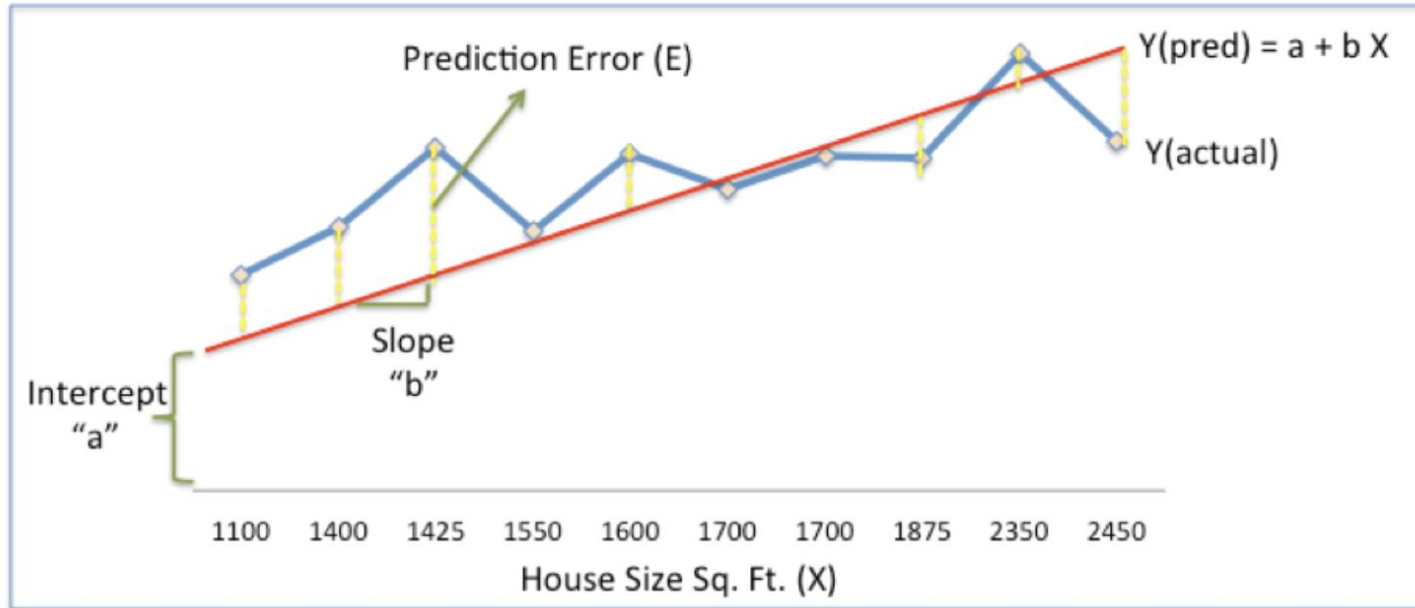
House Size sq.ft (X)	1400	1600	1700	1875	1100	1550	2350	2450	1425	1700
House Price\$ (Y)	245,000	312,000	279,000	308,000	199,000	219,000	405,000	324,000	319,000	255,000

Given its size (X), what will its price (Y) be?





# Gradient Descent: Example



$$\begin{aligned}\text{Sum of Squared Errors (SSE)} &= \frac{1}{2} \text{Sum (Actual House Price - Predicted House Price)}^2 \\ &= \frac{1}{2} \text{Sum}(Y - Y_{\text{pred}})^2\end{aligned}$$

# Gradient Descent: Example

Step 1: Initialize the weights(a & b) with random values and calculate Error (SSE)

Step 2: Calculate the gradient i.e. change in SSE when the weights (a & b) are changed by a very small value from their original randomly initialized value. This helps us move the values of a & b in the direction in which SSE is minimized.

Step 3: Adjust the weights with the gradients to reach the optimal values where SSE is minimized

Step 4: Use the new weights for prediction and to calculate the new SSE

Step 5: Repeat steps 2 and 3 till further adjustments to weights doesn't significantly reduce the Error

# Gradient Descent: Example

HOUSING DATA	
House Size (X)	House Price (Y)
1,100	1,99,000
1,400	2,45,000
1,425	3,19,000
1,550	2,40,000
1,600	3,12,000
1,700	2,79,000
1,700	3,10,000
1,875	3,08,000
2,350	4,05,000
2,450	3,24,000

Normalize

Min-Max Standardization	
X (X-Min/Max-min)	Y (Y-Min/Max-Min)
0.00	0.00
0.22	0.22
0.24	0.58
0.33	0.20
0.37	0.55
0.44	0.39
0.44	0.54
0.57	0.53
0.93	1.00
1.00	0.61

# Gradient Descent: Example

## Step 1

a	b	X	Y	YP=a+bX	SSE=1/2(Y-YP)^2
0.45	0.75	0.00	0.00	0.45	0.101
		0.22	0.22	0.62	0.077
		0.24	0.58	0.63	0.001
		0.33	0.20	0.70	0.125
		0.37	0.55	0.73	0.016
		0.44	0.39	0.78	0.078
		0.44	0.54	0.78	0.030
		0.57	0.53	0.88	0.062
		0.93	1.00	1.14	0.010
		1.00	0.61	1.20	0.176
Total SSE					0.677

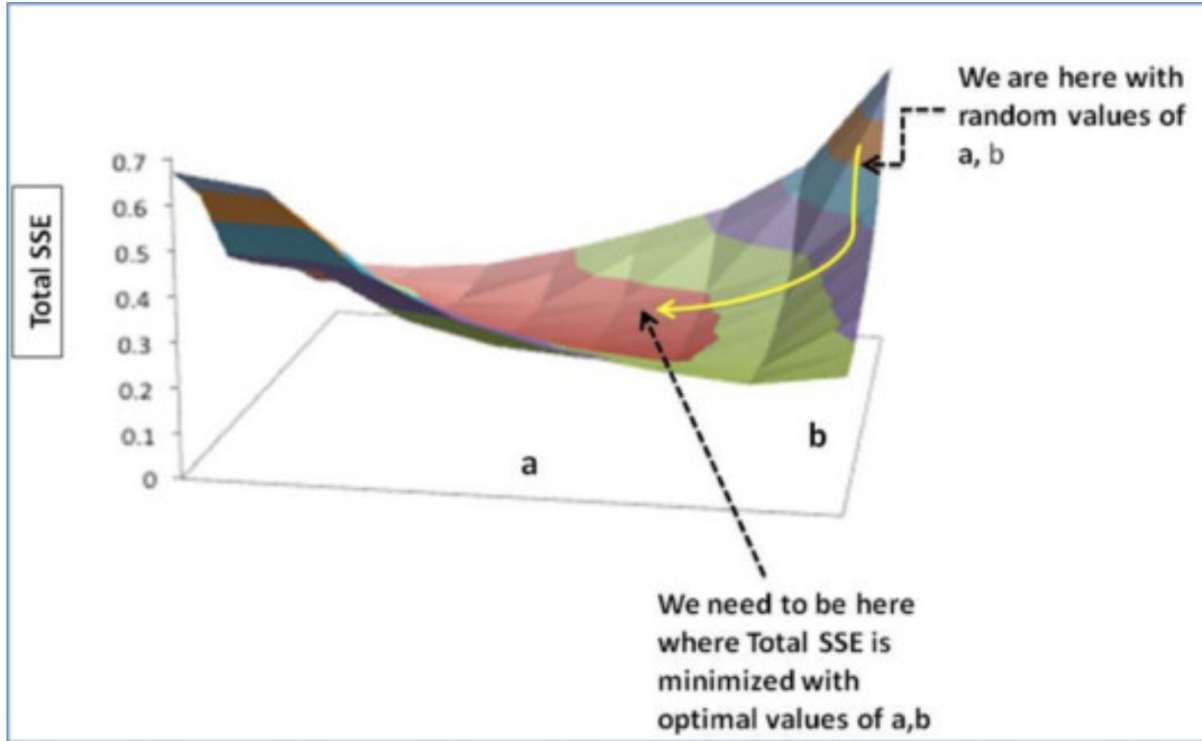
# Gradient Descent: Example

## Step 2

a	b	X	Y	YP=a+bX	SSE	$\partial SSE/\partial a$ = -(Y-YP)	$\partial SSE/\partial b$ = -(Y-YP)X	
0.45	0.75	0.00	0.00	0.45	0.101	0.45	0.00	
		0.22	0.22	0.62	0.077	0.39	0.09	
		0.24	0.58	0.63	0.001	0.05	0.01	
		0.33	0.20	0.70	0.125	0.50	0.17	
		0.37	0.55	0.73	0.016	0.18	0.07	
		0.44	0.39	0.78	0.078	0.39	0.18	
		0.44	0.54	0.78	0.030	0.24	0.11	
		0.57	0.53	0.88	0.062	0.35	0.20	
		0.93	1.00	1.14	0.010	0.14	0.13	
		1.00	0.61	1.20	0.176	0.59	0.59	
Total SSE					0.677	Sum	3.300	1.545

# Gradient Descent: Example

## Step 3



# Gradient Descent: Example

## Step 4

a	b	X	Y	YP=a+bX	SSE	∂SSE/∂a	∂SSE/∂b	
0.42	0.73	0.00	0.00	0.42	0.087	0.42	0.00	
		0.22	0.22	0.58	0.064	0.36	0.08	
		0.24	0.58	0.59	0.000	0.01	0.00	
		0.33	0.20	0.66	0.107	0.46	0.15	
		0.37	0.55	0.69	0.010	0.14	0.05	
		0.44	0.39	0.74	0.063	0.36	0.16	
		0.44	0.54	0.74	0.021	0.20	0.09	
		0.57	0.53	0.84	0.048	0.31	0.18	
		0.93	1.00	1.10	0.005	0.10	0.09	
		1.00	0.61	1.15	0.148	0.54	0.54	
Total SSE					0.553	Sum	2.900	1.350

# Gradient Descent: In depth Analysis

Formula:

$$X = X - lr * \frac{d}{dX} f(X)$$

*Where,*

$X$  = *input*

$F(X)$  = *output based on X*

$lr$  = *learning rate*



# Gradient Descent: Single Variable

Cost Function

$$J(\theta) = \theta^2$$

Goal

$$\min J(\theta)$$

Update Function

$$\theta := \theta - \alpha * \frac{d}{d\theta} J(\theta)$$

Learning Rate

*Learning Rate :*

$$\alpha = 0.1$$

# Gradient Descent: Single Variable

## Updating Parameters

$$\theta := \theta - \alpha * \frac{d}{d\theta} J(\theta)$$

$$\theta := \theta - \alpha * 2\theta$$

$$\theta := \theta - 2\alpha\theta$$

$$\theta := 0.8 * \theta$$

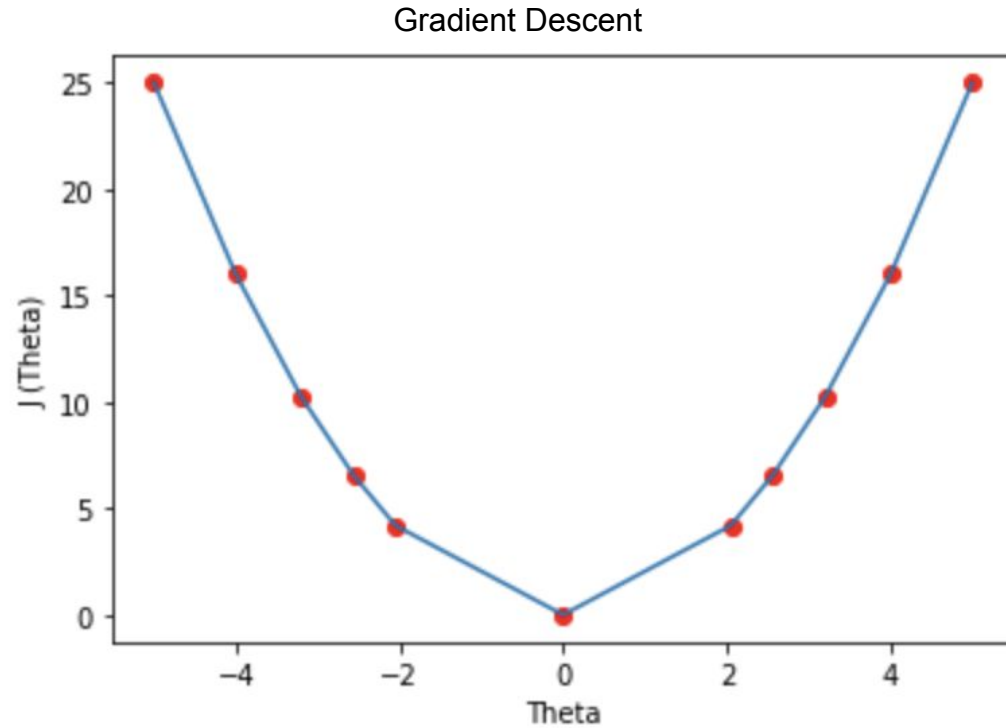
## Table Generation

# Gradient Descent: Single Variable

$\theta$	$J(\theta)$
5	25
4	16
3.2	10.24
2.56	6.55
2.04	4.19
0	0

$\theta$	$J(\theta)$
-5	25
-4	16
-3.2	10.24
-2.56	6.55
-2.04	4.19
0	0

# Gradient Descent: Single Variable



# Gradient Descent: Multiple Variables

Cost Function

$$J(\theta_1, \theta_2) = \theta_1^2 + \theta_2^2$$

Goal

$$\min J(\theta_1, \theta_2)$$

Update Function

$$\theta_1 := \theta_1 - \alpha * \frac{d}{d\theta_1} J(\theta_1, \theta_2)$$

$$\theta_2 := \theta_2 - \alpha * \frac{d}{d\theta_2} J(\theta_1, \theta_2)$$

# Gradient Descent: Multiple Variables

## Derivatives

$$\begin{aligned}\frac{d}{d\theta_1}J(\theta_1, \theta_2) &= \frac{d}{d\theta_1}(\theta_1^2 + \theta_2^2) \\ &= \frac{d}{d\theta_1}(\theta_1^2) + \frac{d}{d\theta_1}(\theta_2^2) \\ &= 2\theta_1 + 0 \\ &= 2\theta_1\end{aligned}$$

$$\begin{aligned}\frac{d}{d\theta_2}J(\theta_1, \theta_2) &= \frac{d}{d\theta_2}(\theta_1^2 + \theta_2^2) \\ &= \frac{d}{d\theta_2}(\theta_1^2) + \frac{d}{d\theta_2}(\theta_2^2) \\ &= 0 + 2\theta_2 \\ &= 2\theta_2\end{aligned}$$

# Gradient Descent: Multiple Variables

## Update Values

$$\theta_1 := \theta_1 - \alpha * 2\theta_1$$

$$\theta_1 := \theta_1 - 2\alpha\theta_1$$

$$\theta_2 := \theta_2 - \alpha * 2\theta_2$$

$$\theta_2 := \theta_2 - 2\alpha\theta_2$$

## Learning Rate

*Learning Rate :*

$$\alpha = 0.1$$

# Gradient Descent: Multiple Variables

Table

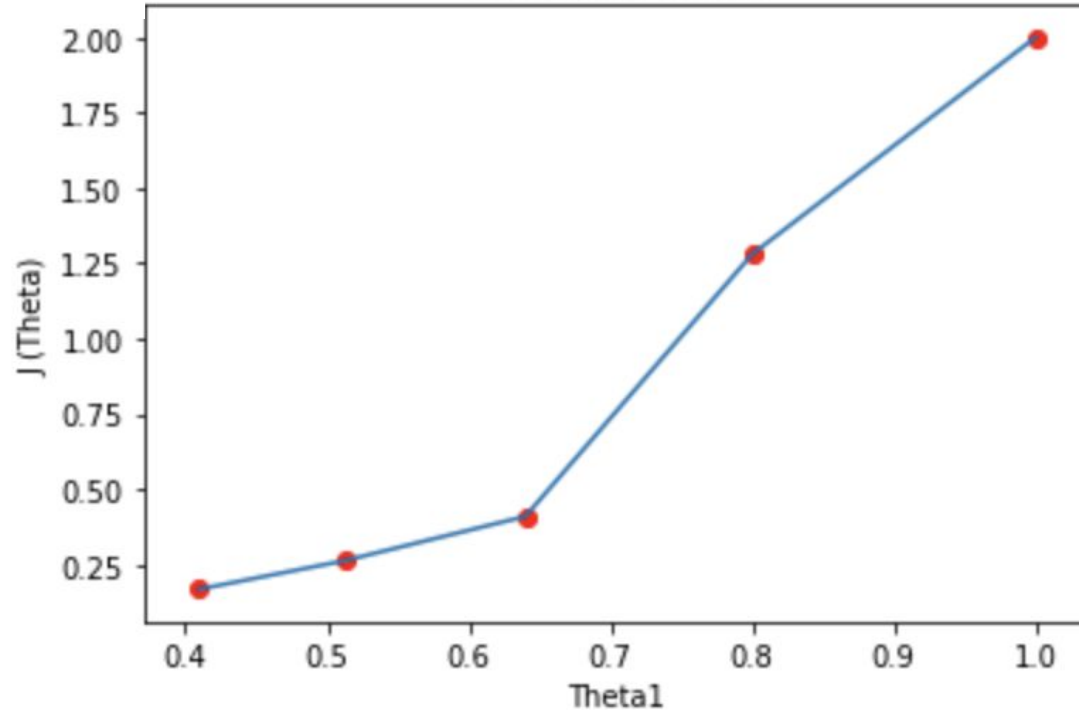
$\theta_1$	$\theta_2$	$J(\theta)$
1	1	2
0.8	0.8	1.28
0.64	0.64	0.4096
0.512	0.512	0.2621
0.4096	0.4096	0.1677
0	0	0



# Gradient Descent: Multiple Variables

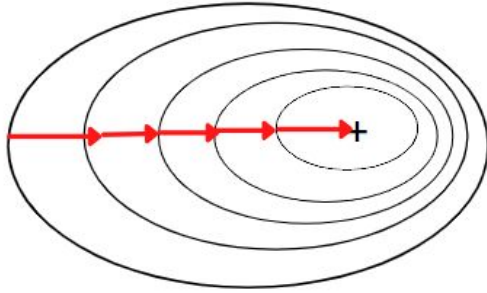
Gradient Descent

Graph

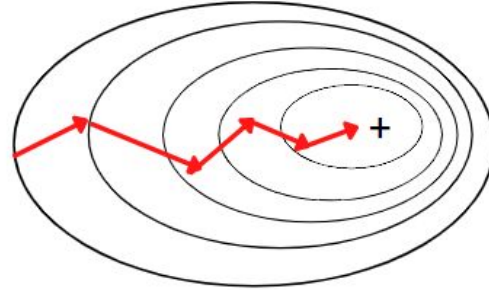


# Gradient Descent: Types

**Batch Gradient Descent**



**Mini-Batch Gradient Descent**



**Stochastic Gradient Descent**

