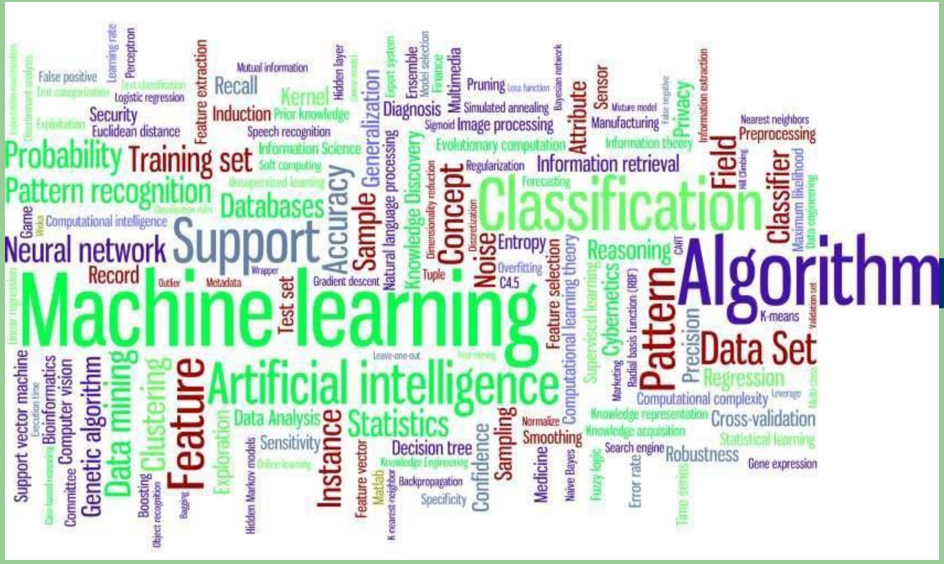
Machine Learning with Python



Lecture 2

Regression

الانحدار (التوقع)

Types of Machine Learning

- **Supervised (inductive) learning**

  Given: training data + desired outputs (labels)
- **Unsupervised learning**

  Given: training data (without desired outputs)
- **Semi-supervised learning**

  Given: training data + a few desired outputs
- **Reinforcement learning**

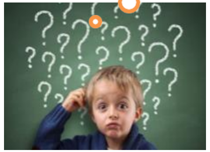
  Rewards from sequence of actions

Supervised Learning

Supervised → **Uses Labeled data**

- **What is this?**

What's this?

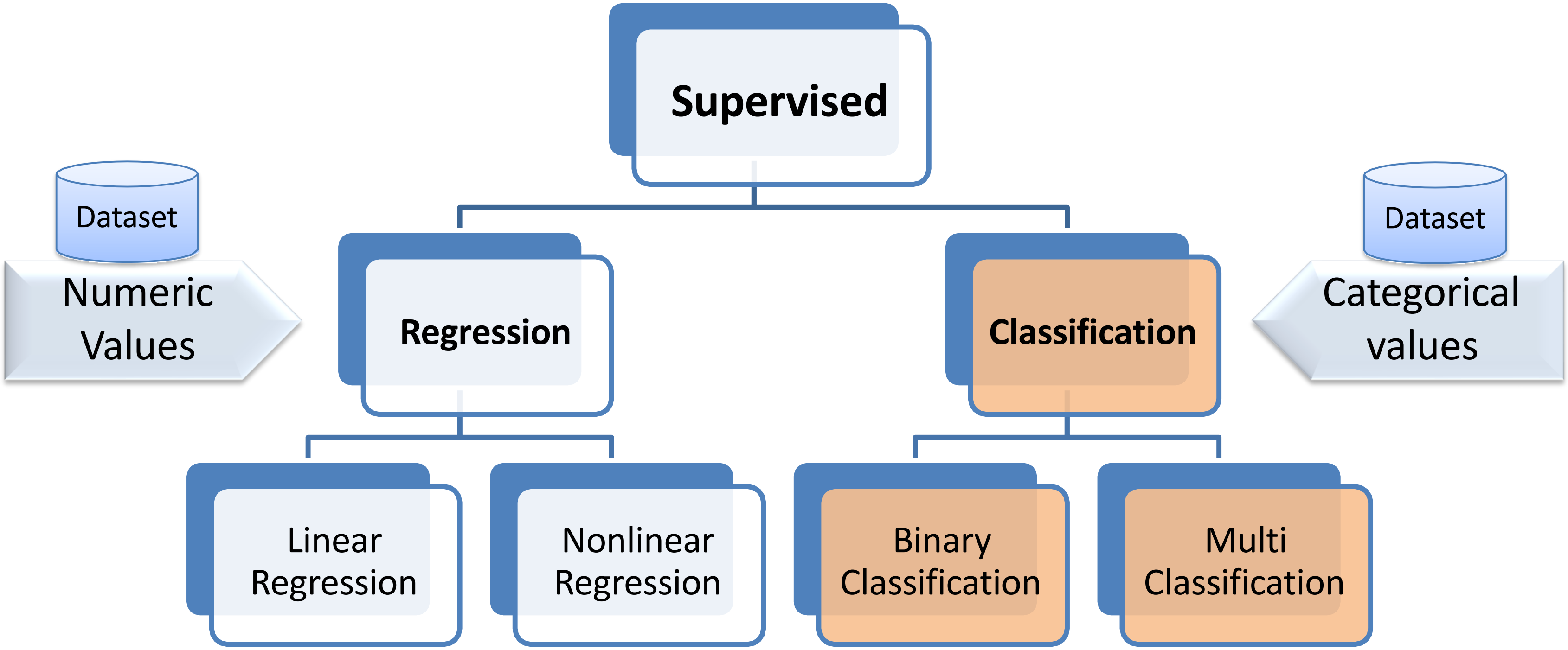


Features **Label**

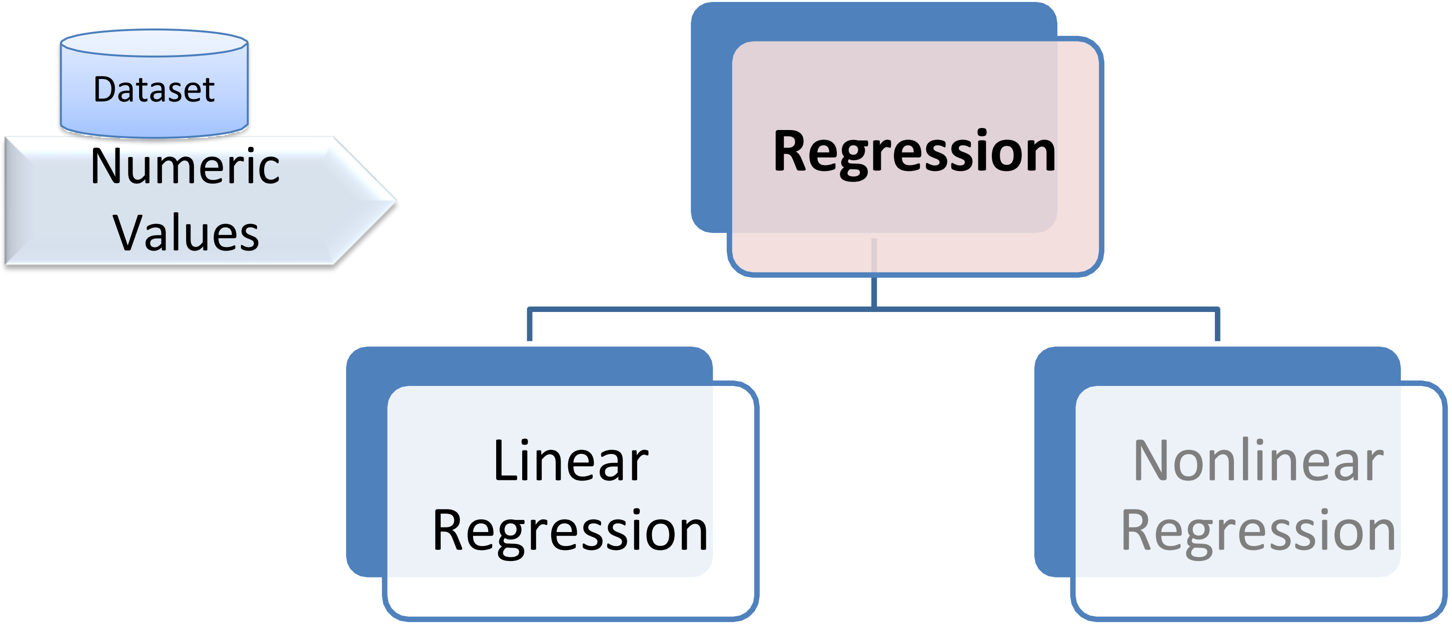
(big ears, big nose, big mouth, DOG)

Classify or Predict target variables

Supervised Learning



Supervised Learning



What is a Regression?

Regression

- Regression problem: the output (y) that you want is continues value (numeric). *(e.g., a rating: a real number between 0-10, # of followers, house price)*

❑ *Predicting* continuous outputs is called ***regression***

What do I need in order to predict these outputs?

- Features (inputs), we'll call these **x** (or *x* if vectors)
- Training examples, many $x^{(i)}$ for which $t^{(i)}$ is known (e.g., many movies for which we know the rating)
- A model, a function that represents the relationship between **x** and **t**
- A loss or a cost or an objective function, which tells us how well our model approximates the training examples
- Optimization, a way of finding the parameters of our model that minimizes the loss function

What is a Regression?

Regression: is a statistical approach for modeling the relationship between some variables x (features) and some real valued outcome y (target).

x y

Data point → $(2, 4)$

$(1, 2)$

$(0, 0)$

$(3, ??)$

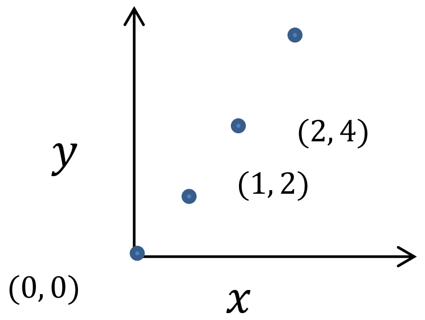
Model

$f(x) = 2x$

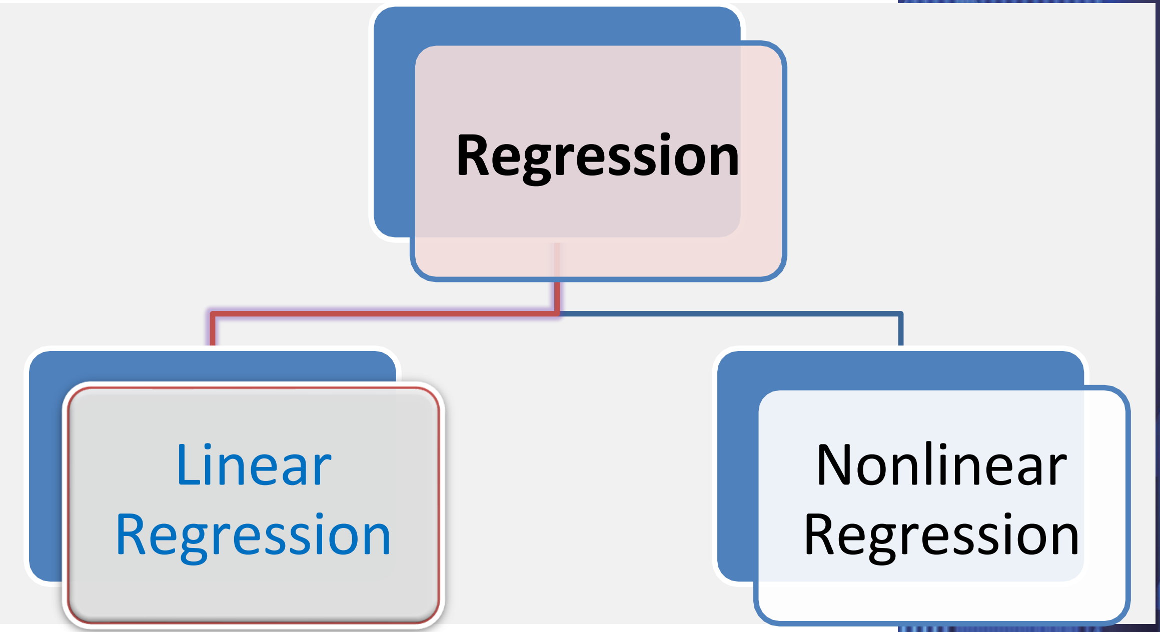
$f(x) = 2 * 3$

$(3, 6)$

Cartesian drawing



Linear Regression



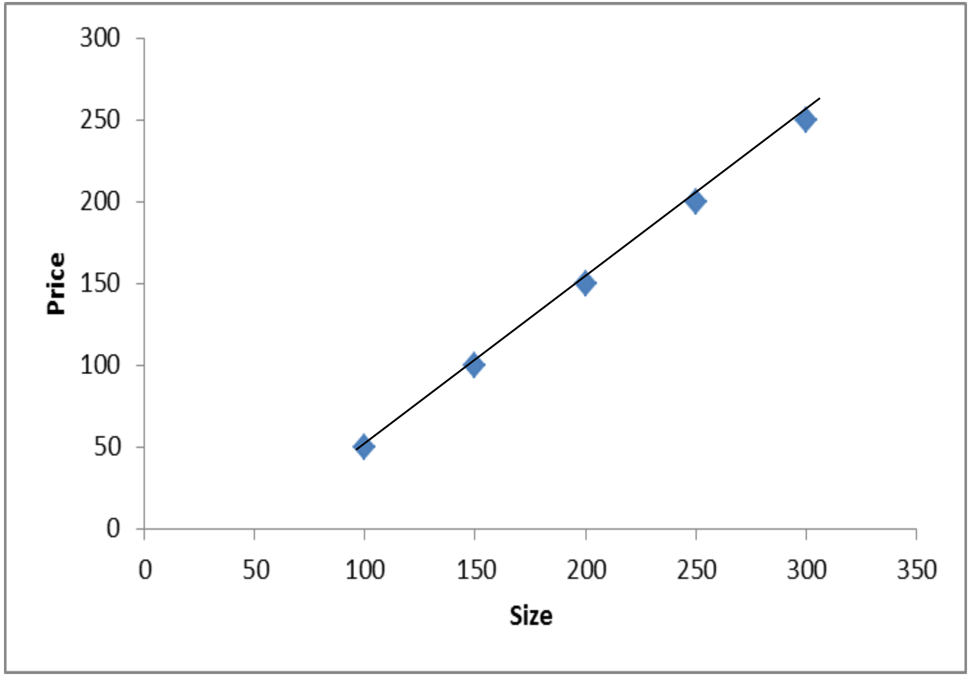
Linear Regression

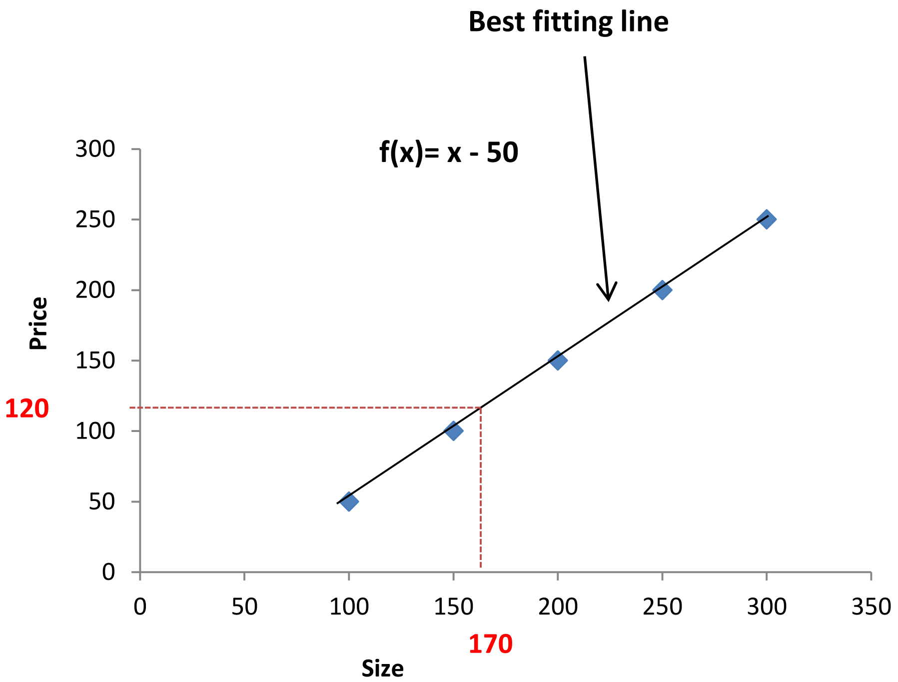
Let's say we have this dataset.

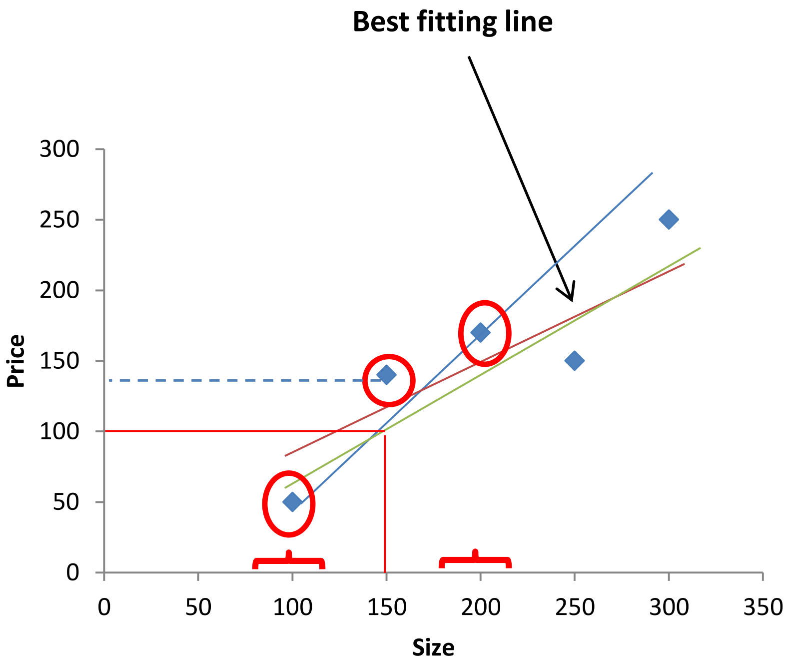
x → Size

y → Price

Size	Price
100	50
150	100
200	150
250	200
300	250







- **Key Questions:**
  - How do we parametrize the model?
  - What loss (objective) function should we use to judge the t?
  - How do we optimize (generalization)?

Linear Function

$$\boldsymbol{f(x) = \theta x + b}$$

$f(x) = model \rightarrow \boxed{y}$

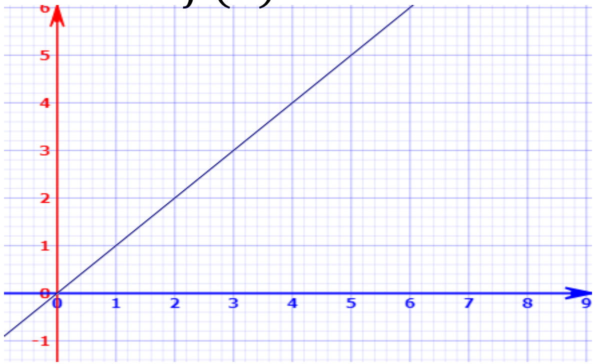
$\boldsymbol{x}$: ***variable (feature)***

$\boldsymbol{\theta}$: **coefficient (Slope) / الميل**

$\boldsymbol{b}$: **y axis-intercept / Bias**

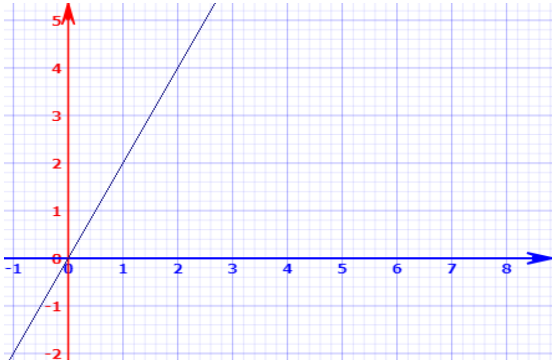
$$\boxed{\begin{aligned}\theta &= 1\\ b &= 0\end{aligned}}$$

$f(x) = x$



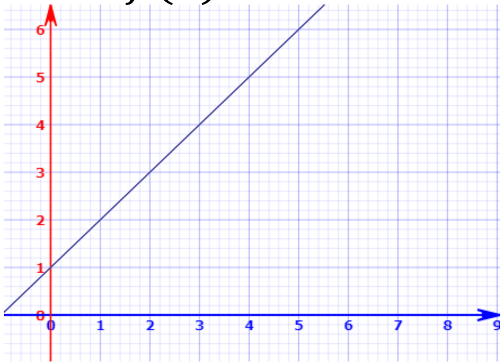
$$\boxed{\begin{aligned}\theta &= 2\\ b &= 0\end{aligned}}$$

$f(x) = 2x$



$$\boxed{\begin{aligned}\theta &= 1\\ b &= 1\end{aligned}}$$

$f(x) = x + 1$

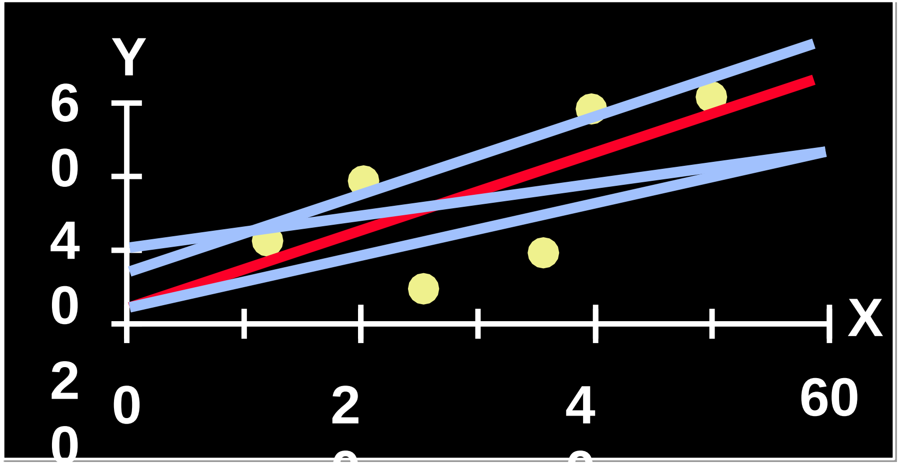


Scatter plot

- Plot of All (X_i, Y_i) Pairs
- Suggests How Well Model Will Fit

How would you draw a line through the points?
How do you determine which line 'fits best'? →

- ✓ Slope changed
- ✓ Intercept changed



Let's say we have:

Unseen

Dataset

Features →

x	-1	0	1	2	3	4	5
y	-1	1	3	5	7	9	?

Labels →

Target

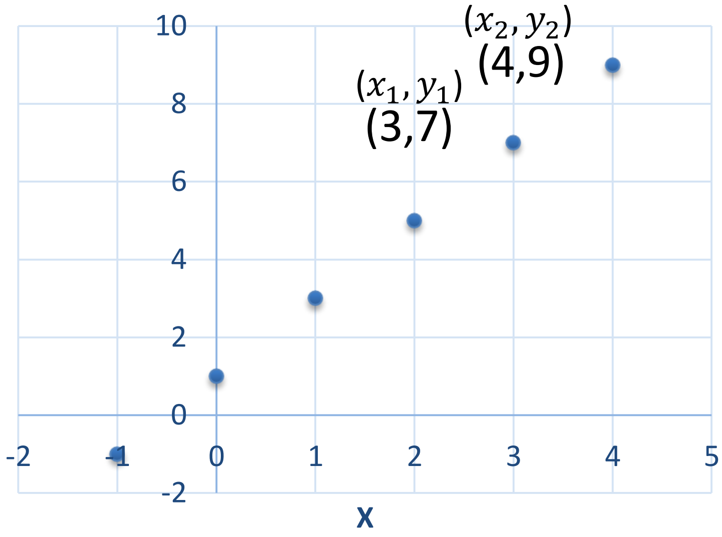
$$f(x) = ???$$

x	-1	0	1	2	3	4
y	-1	1	3	5	7	9

Steps:

1. Visualize
2. Find parameter θ
3. Find parameter b

Step 1:



Step 2: θ is slope

Step 3:

$$\theta = \frac{y_2 - y_1}{x_2 - x_1}$$

$$\theta = \frac{9-7}{4-3}$$

$$\boxed{\theta = 2}$$

$$\theta = f(x)'$$

$$f(x)' = 2$$

$$\int f(x)' = 2x + b$$

$$b = \bar{y} - \theta\,\bar{x}$$

$$b = \frac{24}{6} - 2\frac{9}{6}$$

$$\boxed{b=1}$$

$$\boxed{f(x) = 2x + 1}$$

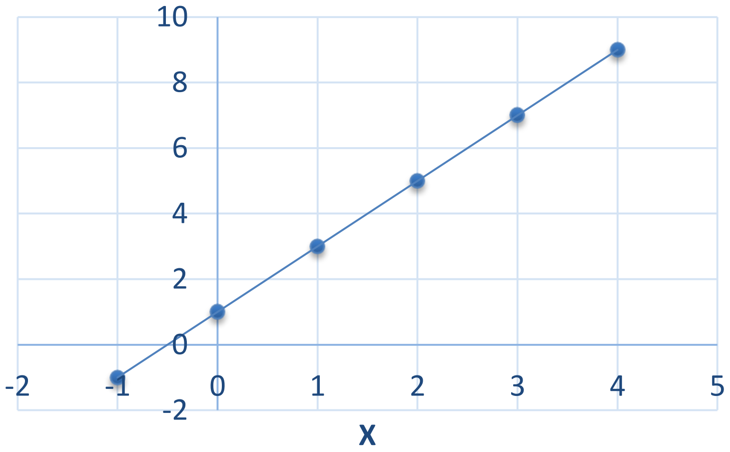
Let’s say we have:

Dataset

Unseen

Features	x	-1	0	1	2	3	4	5
Labels	y	-1	1	3	5	7	9	11

Target



$$\boxed{f(x) = 2x + 1}$$

A model to predict new variables is required

$$\boldsymbol{f(x) = \theta x + b}$$

$y \rightarrow output$\target value

$$h(x) = \theta_i x_i + \theta_0$$

Model

Parameters or weights

Features

$(x, y) \rightarrow datapoint$/training example

Linear Regression → (x, y) → Supervised Learning

Feature

Label

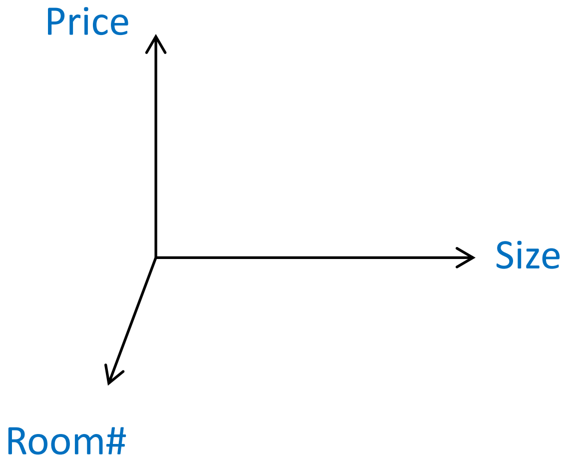
Add a new feature:

Size		Room#	PricePrice	
100		4	60006000	
130		5	70007000	
170		5	50005000	
190		6	90009000	
200		6	1100011000	

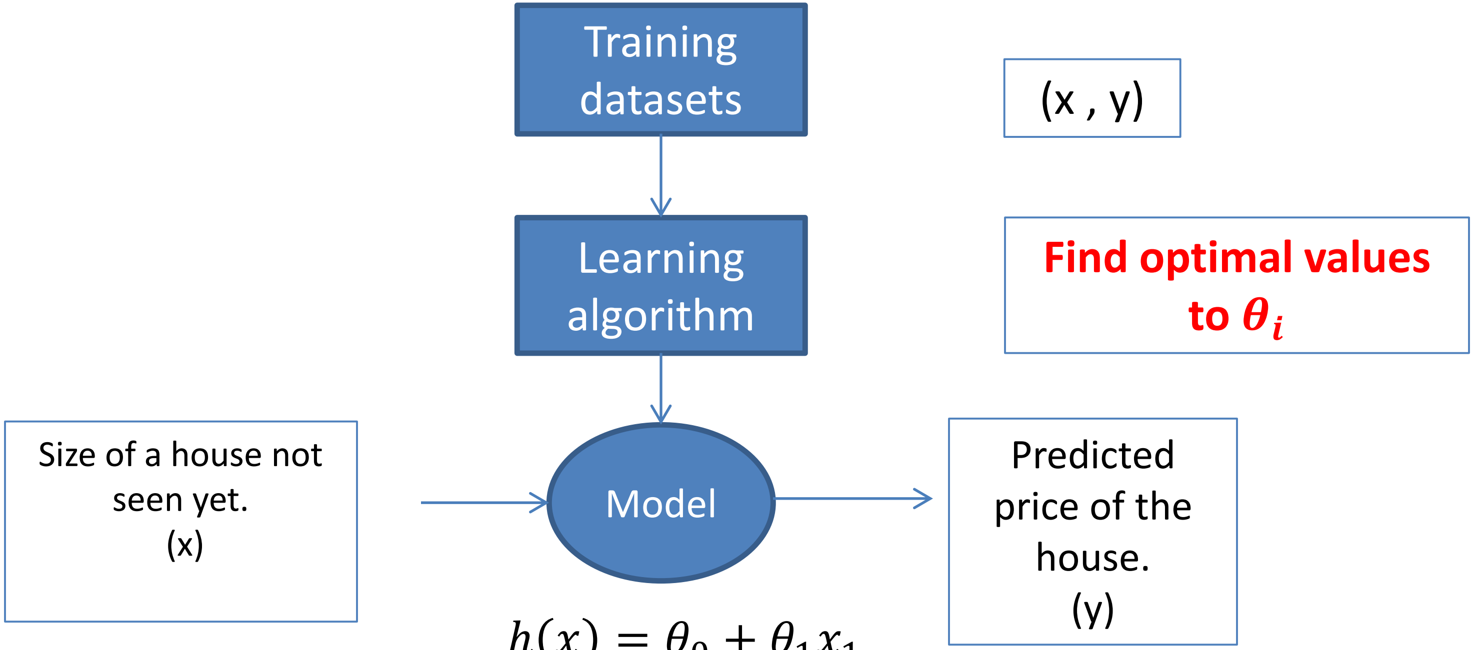
x_1 x_2 y

$$h(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

Find $\theta_0, \theta_1, \theta_2$



Structure of Supervised Learning



$$h(x) = \theta_0 + \theta_1 x_1$$

$n\ features$ $$h(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_j x_j$$ Where x_0 =1

Let's simplify $$h(x) = \sum_{j=0}^{n} \theta_j x_j$$

n= # of features
$j = j^{th}$, counter

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ . \\ . \\ . \\ \theta_n \end{bmatrix} \qquad x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ . \\ . \\ . \\ x_n \end{bmatrix}$$

The job of the learning algorithm is to find values of θ_n such that $\boldsymbol{h(x) \approx y}$

Finding θ_n

Size	Room#	Price
100	4	6000

$$y = \theta_0 + \theta_1 100 + \theta_2 4$$

Initialization

$$\theta_n \ (\theta_0 = 50, \theta_1 = 50, \theta_2 = 100)$$

$$h(x) = 5450$$

Calculate the squared difference between the actual value **y** and the predicted value **h(x)**

$$\boxed{(h(x) - y)^2}$$

$$(5450 - 6000)^2 = 302500$$

Our objective to choose values of θ to **minimize** this **cost** (error) **function**

$$\boxed{min(y - h(x))^2}$$

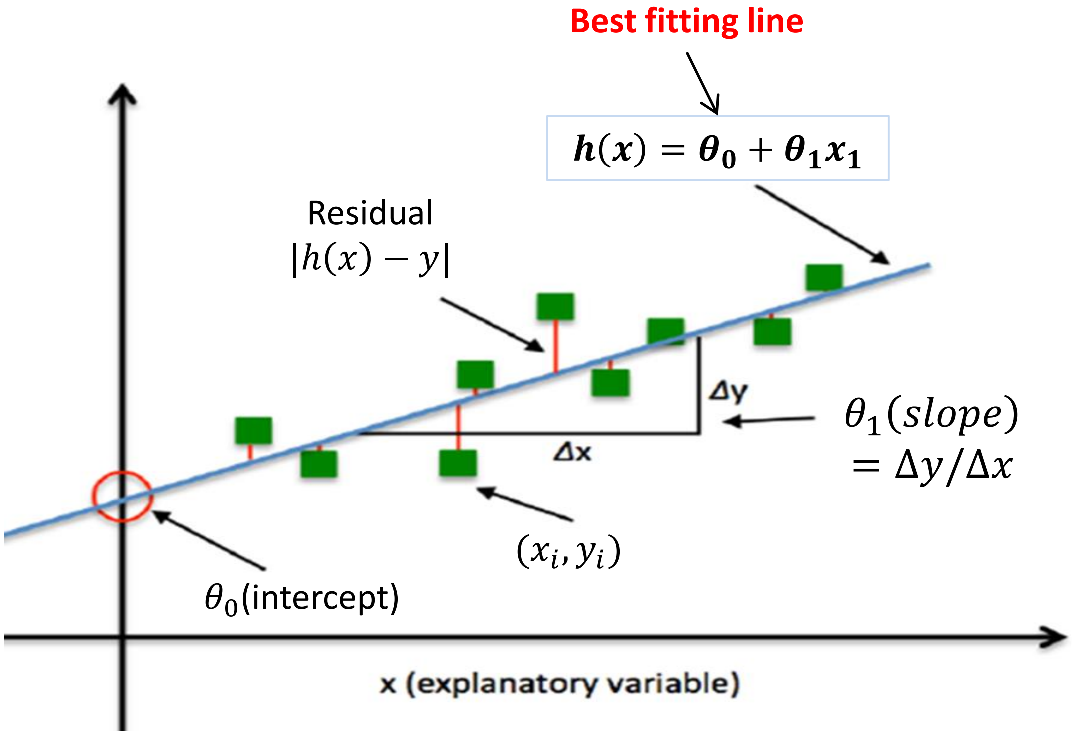
Least Squares Error

'**Best Fit**' Means Difference Between Actual Y Values & Predicted Values Are a Minimum.

$$\boxed{\boldsymbol{min(h(x) - y)^2}}$$

- Sum of Squared errors (Differences) (SSE))

$$\textbf{SSE} = \sum_{i=1}^{m}\left(h\left(x^{(i)}\right) - y^{(i)}\right)^2$$



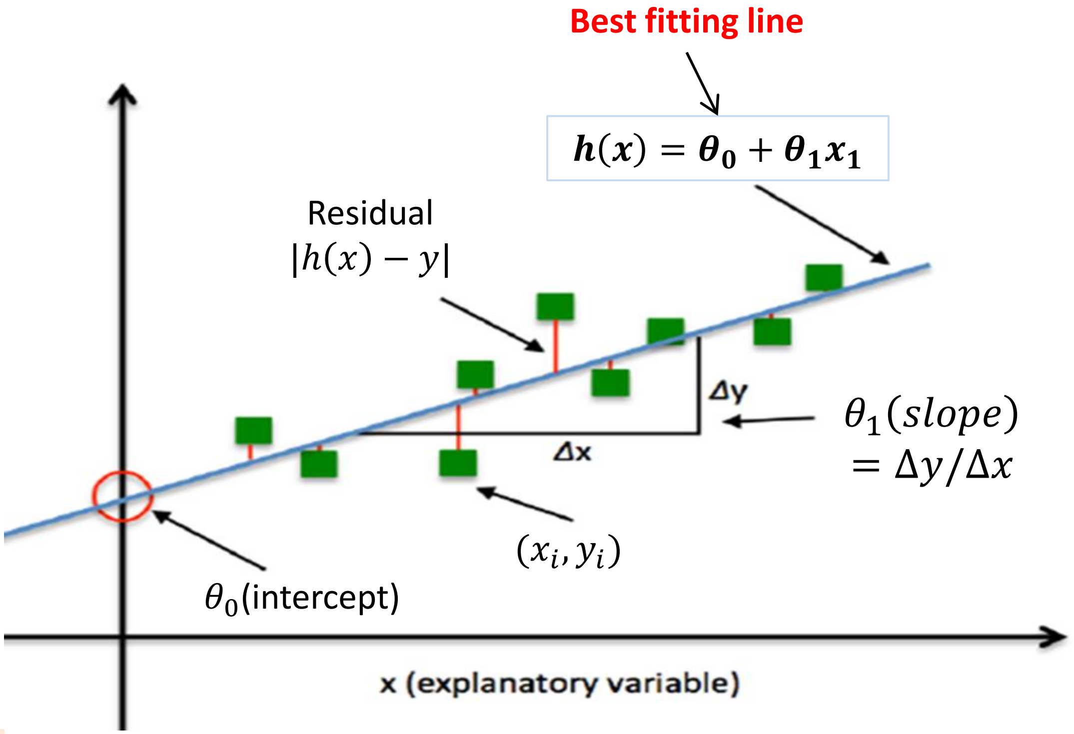
Least Squares Error

- Sum of Squared Errors (Residual) (SSE))

$$\textbf{SSE} = \sum_{i=1}^{m} \left(h\left(x^{(i)}\right) - y^{(i)}\right)^2$$

- **Least Squares (LS)** is the Minimizes the Sum of Squared errors

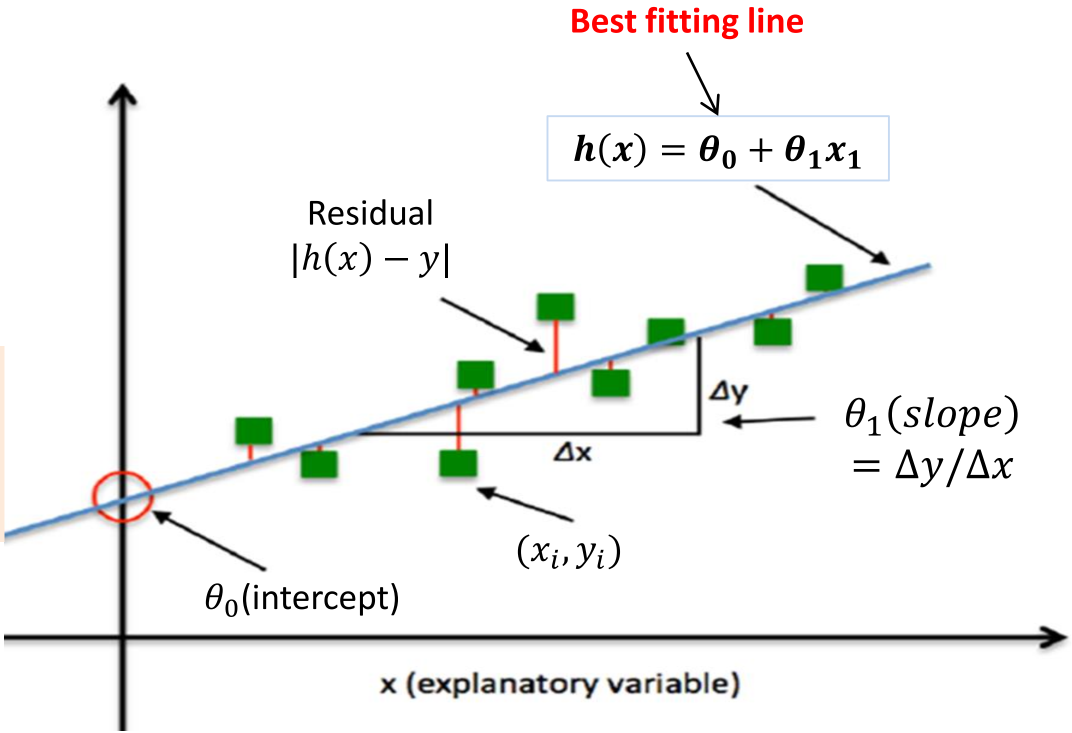
$$\mathbf{LS} = \min \sum_{i=1}^{m} \left(h\left(x^{(i)}\right) - y^{(i)}\right)^2$$



Least Squares Error

- **Least Squares (LS)** is the Minimizes the Sum of Squared errors

$$\textbf{LS} = \min \sum_{i=1}^{m} \left(h\left(x^{(i)}\right) - y^{(i)}\right)^2$$



How we can calculate the <u>min</u>?

- **Mean Absolute Error (MAE)**
- **Root Mean Squared Error (RMSE)**
- **Relative Absolute Error (RAE)**

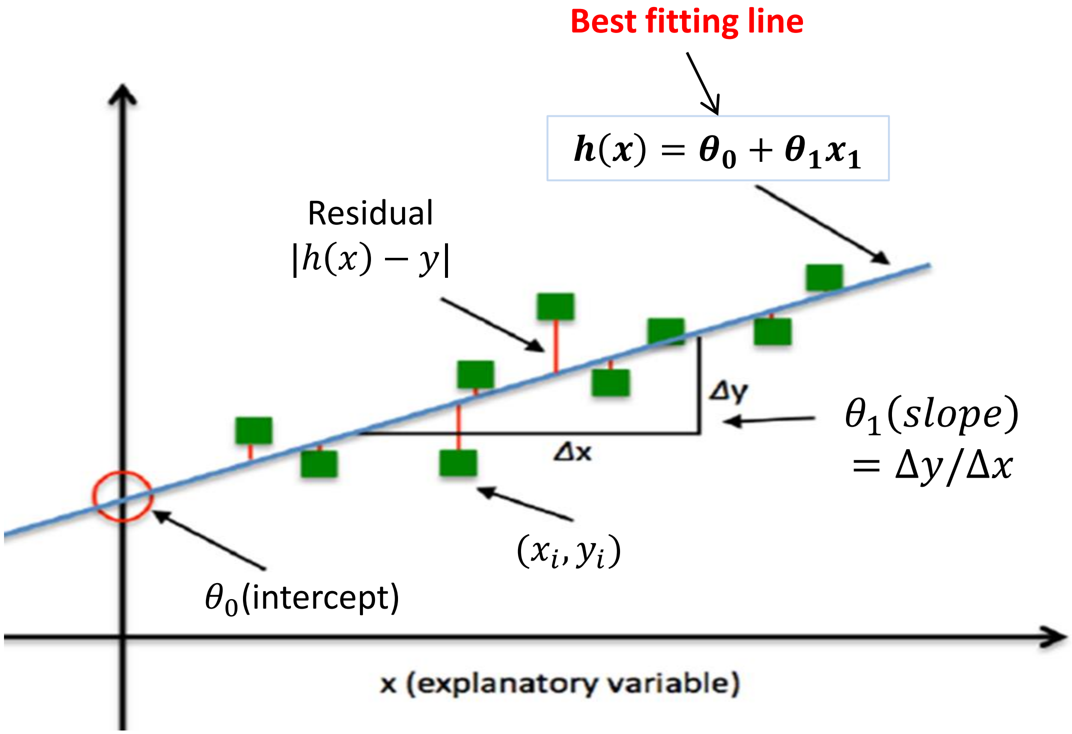
Least Squares Error

- **Least Squares (LS)**

$$\mathbf{LS} = \min \sum_{i=1}^{m} (h(x^{(i)}) - y^{(i)})^2$$

This cost function is called Ordinary Least Squares, and defined as follows:

$$j(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h(x^{(i)}) - y^{(i)})^2$$



The **goal** of learning is to **minimize the cost function** $j(\theta)$.

Cost function visualization

Consider a simple case of hypothesis by setting θ_0=0, then ***h*** becomes : $\mathbf{h_\theta(x)=\theta_1 x}$

Each value of θ_1 corresponds to a different hypothesis as it is the **slope** of the line

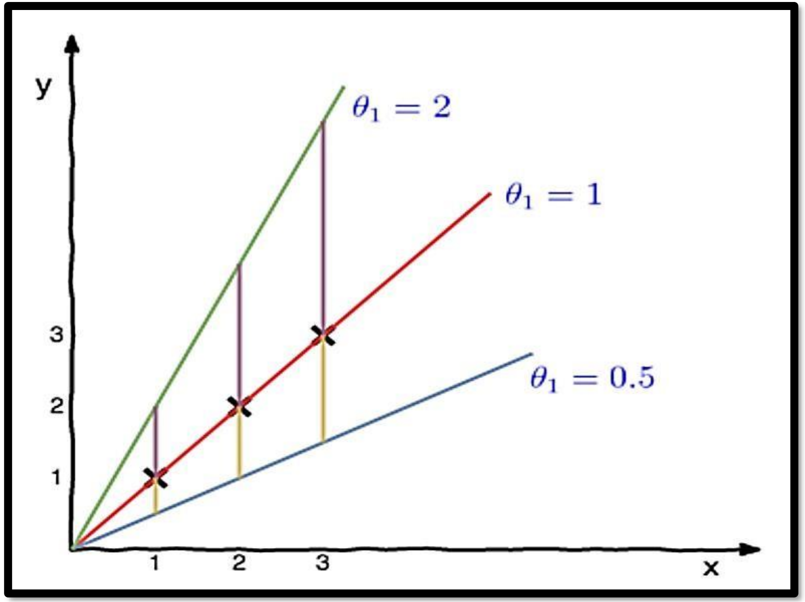
which corresponds to different lines passing through the **origin** as shown in plots below as **y-intercept** i.e. θ_0 is nulled out.

$$J(\theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(\theta_1 x^{(i)} - y^{(i)}\right)^2$$

At θ_1=2, $J(2) = \frac{1}{2*3}(1^2+2^2+3^2) = \frac{14}{6} = 2.33$

At θ_1=1, $J(1) = \frac{1}{2*3}(0^2+0^2+0^2) = 0$

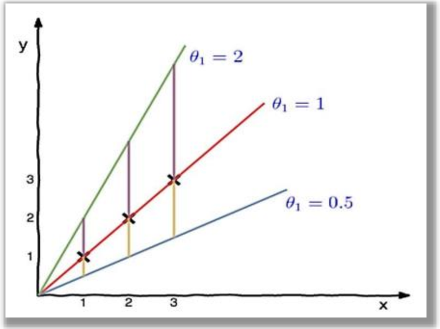
At θ_1=0.5, $J(0 = \frac{1}{2*3}(0.5^2+1^2+1.5^2) = 0.58$



Simple Hypothesis

Cost function visualization

$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(\theta_1 x^{(i)} - y^{(i)}\right)^2$$

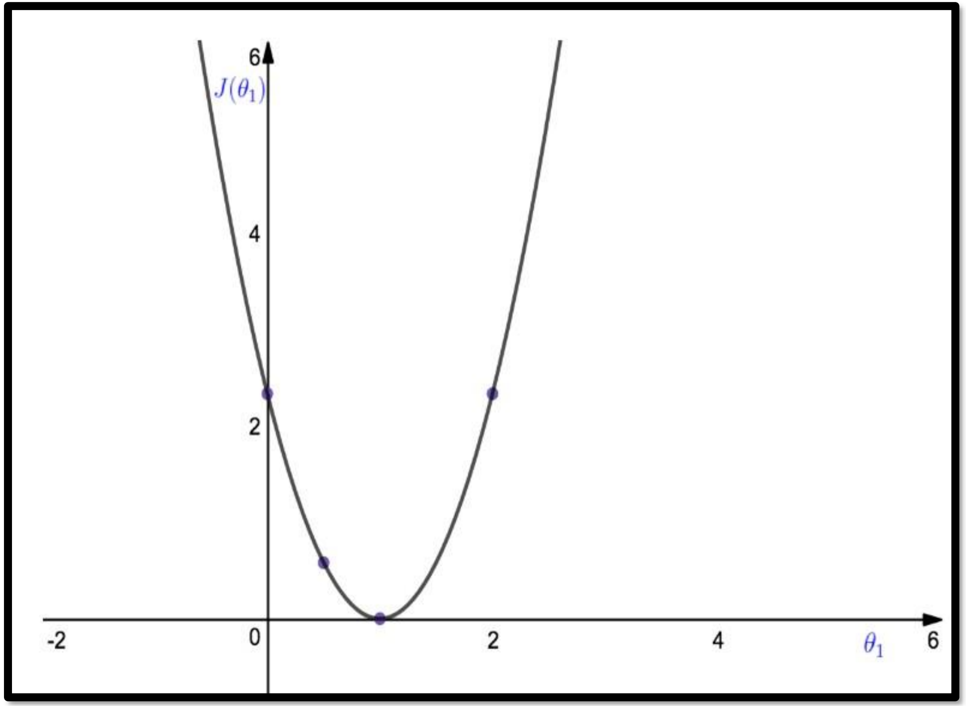


At θ_1=2, $J(2) = \frac{1}{2*3}(1^2 + 2^2 + 3^2) = \frac{14}{6} = 2.33$

At θ_1=1, $J(1) = \frac{1}{2*3}(0^2 + 0^2 + 0^2) = 0$

At θ_1=0.5, $J(0.5) = \frac{1}{2*3}(0.5^2 + 1^2 + 1.5^2) = 0.58$

On **plotting points** like this further, one gets the following graph for the cost function which is dependent on parameter θ_1.



plot each value of θ_1 corresponds to a different hypothesizes

Cost function visualization

$$J(\theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(\theta_1 x^{(i)} - y^{(i)}\right)^2$$

What is the optimal value of θ_1 that minimizes $J(\theta_1)$?

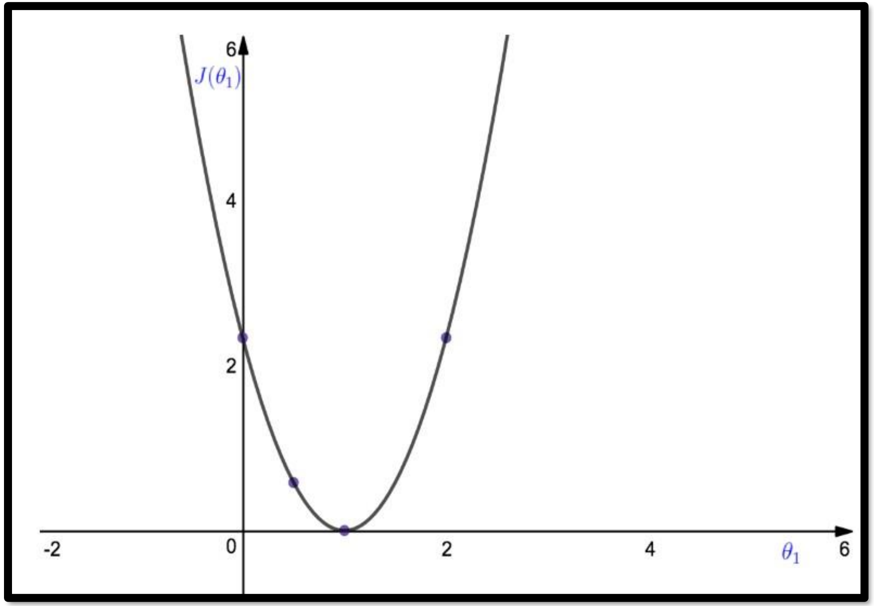
It is clear that best value for $\theta_1 = 1$ as $J(\theta_1) = 0$, which is the minimum.

How to find the best value for θ_1 ?

Plotting ?? Not practical specially in high dimensions?

The solution :

- **Analytical solution: not applicable for large datasets**
- **Numerical solution: ex: Gradient Descent .**



- Next …

Gradient Descent ==> To minimize the cost function $j(\theta)$.