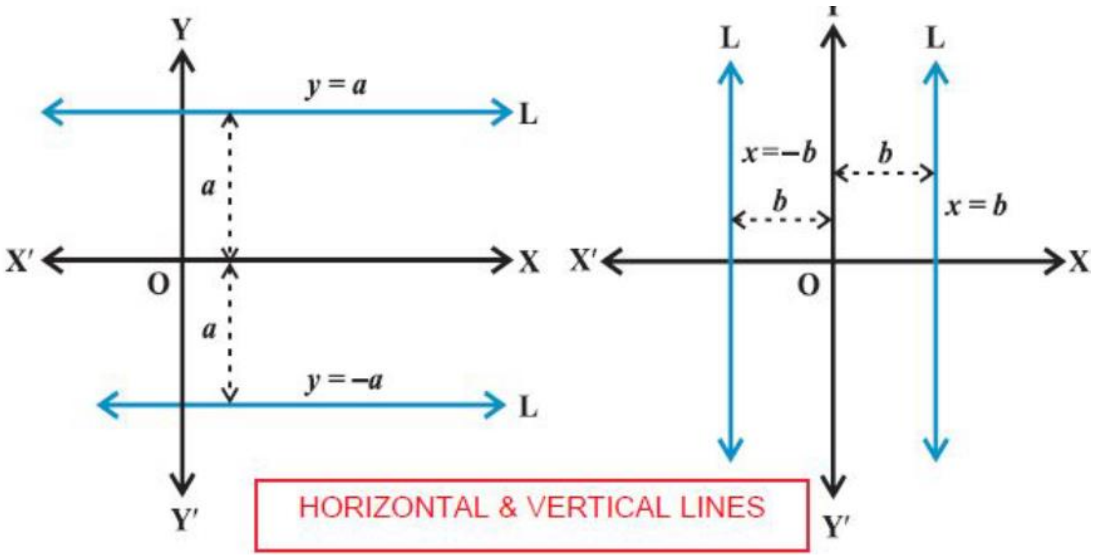
Linear Regression & Gradient Descent With Quick Review of Math involved

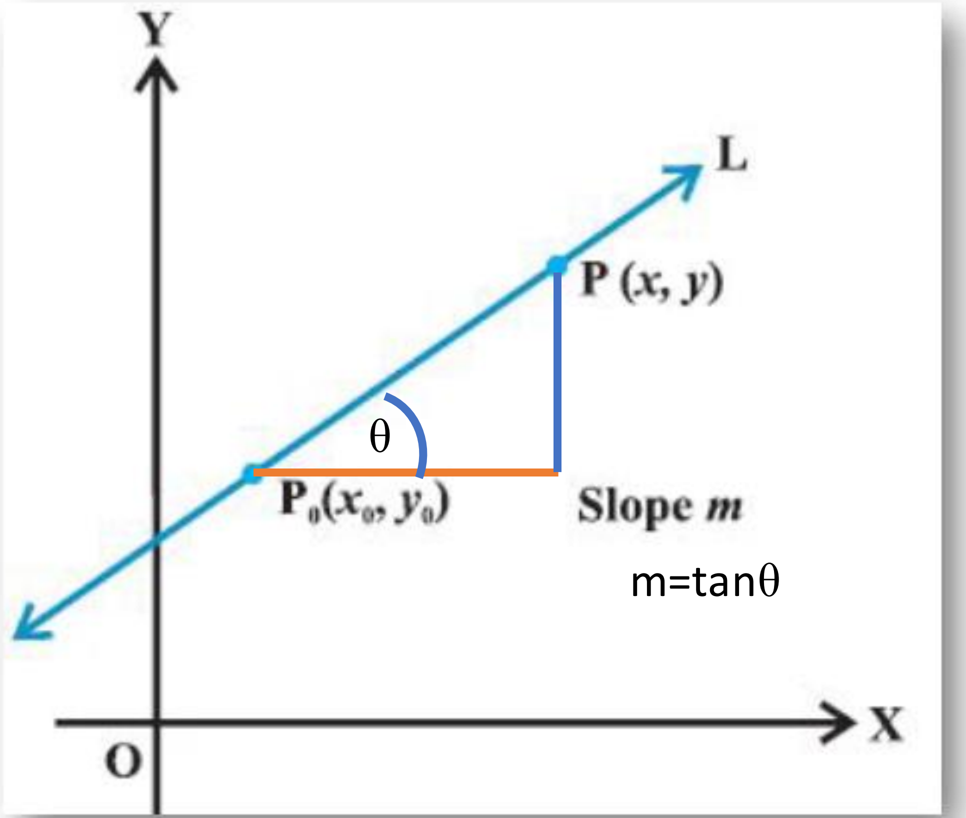
Ramendra Kumar

Reference: http://cs229.stanford.edu

Equations of Straight line

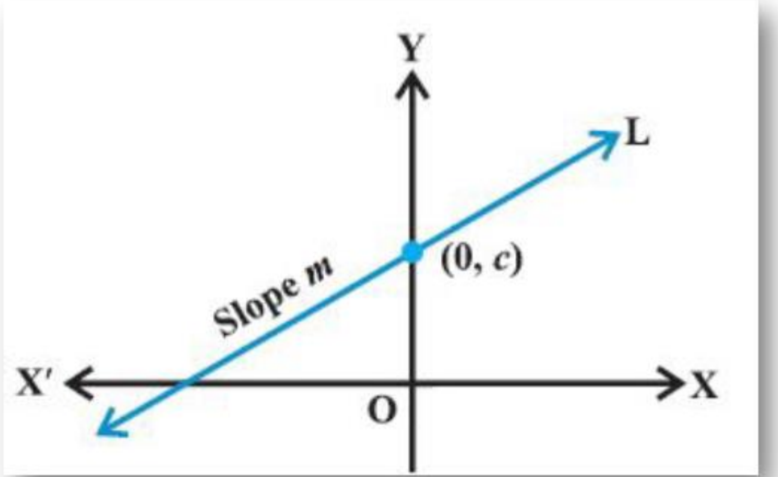


Point Slope Form



$$m = \frac{y - y_0}{x - x_0}, \text{ i.e., } y - y_0 = m(x - x_0)$$

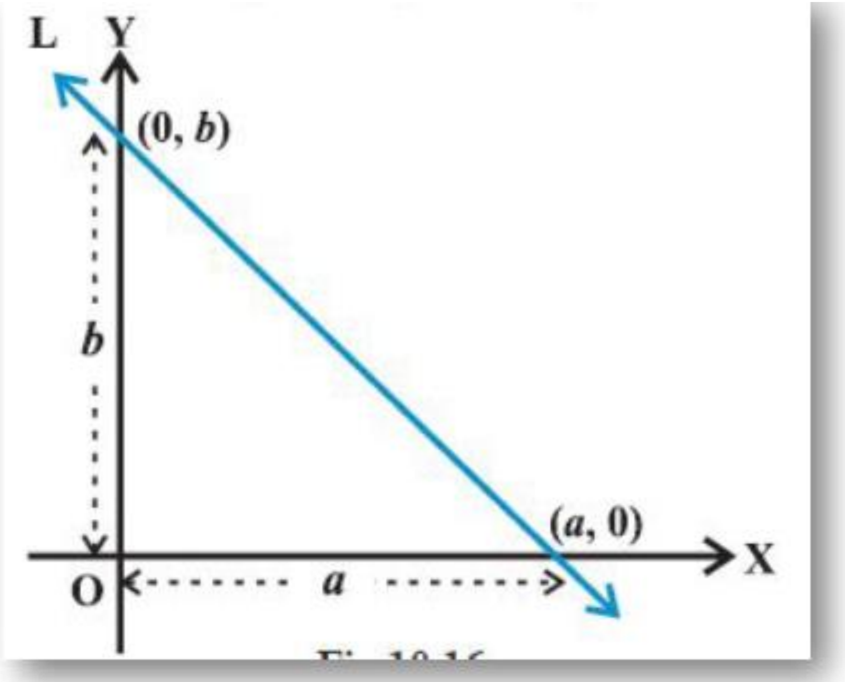
Slope Intercept Form



$$y = mx + c$$

$$y = m(x - d)$$

Intercept Form



$$\frac{x}{a} + \frac{y}{b} = 1.$$

Straight line equations of different forms

$y = mx + c$ → Gradient-intercept form

$y - y_1 = m(x - x_1)$ → *Given gradient and 1 point

$\frac{x}{a} + \frac{y}{b} = 1$ → Double-intercept form

$ax + by + c = 0$ → General form

$\Rightarrow ax + by + c_1 = 0$ → 3x+4y+8=0

or, $by = -ax + (-c_1)$

$or, \quad y = \left(-\frac{a}{b}\right)x + \left(-\frac{c_1}{b}\right)$

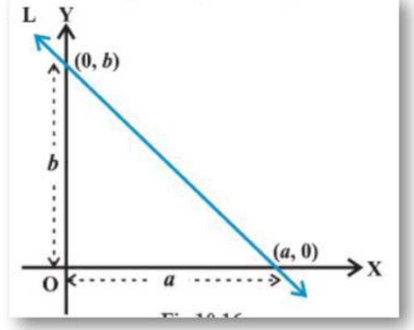
$\rightarrow m = \left(-\frac{a}{b}\right)$ $and\ c = \left(-\frac{c_1}{b}\right)$

$\Rightarrow a_1 x + b_1 y + c_1 = 0$

$or, a_1 x + b_1 y = -c_1$

or, $\left(\frac{a_1}{-c_1}\right)x + \left(\frac{b_1}{-c_1}\right)y = 1$

or, $\left(\frac{x}{\frac{-c_1}{a_1}}\right) + \left(\frac{y}{\frac{-c_1}{b_1}}\right) = 1$



$\frac{x}{a} + \frac{y}{b} = 1.$

$\rightarrow$ a= $\frac{-c_1}{a_1}$ $and\ b = \left(-\frac{c_1}{b_1}\right)$

Draw the graph of $y = 3x + 1$ for values of x from -3 to 3.

x	-3	-2	-1	0	1	2	3
$y = 3x + 1$	-8	-5	-2	1	4	7	10
	↓	↓	↓	↓	↓	↓	↓
	$(-3, -8)$	$(-2, -5)$	$(-1, -2)$	$(0, 1)$	$(1, 4)$	$(2, 7)$	$(3, 10)$

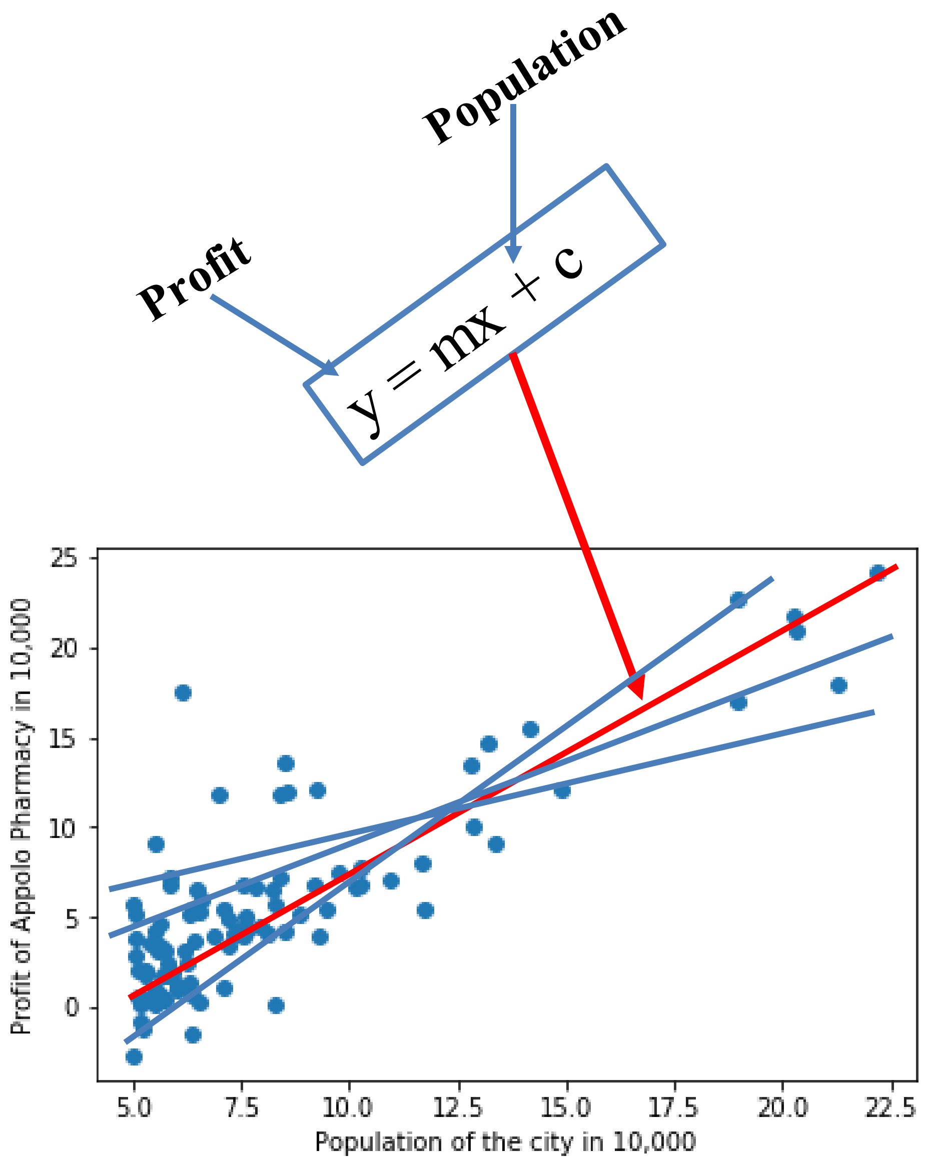
This equation means that we need to multiply every x value by 3 then add 1 to find the corresponding y value

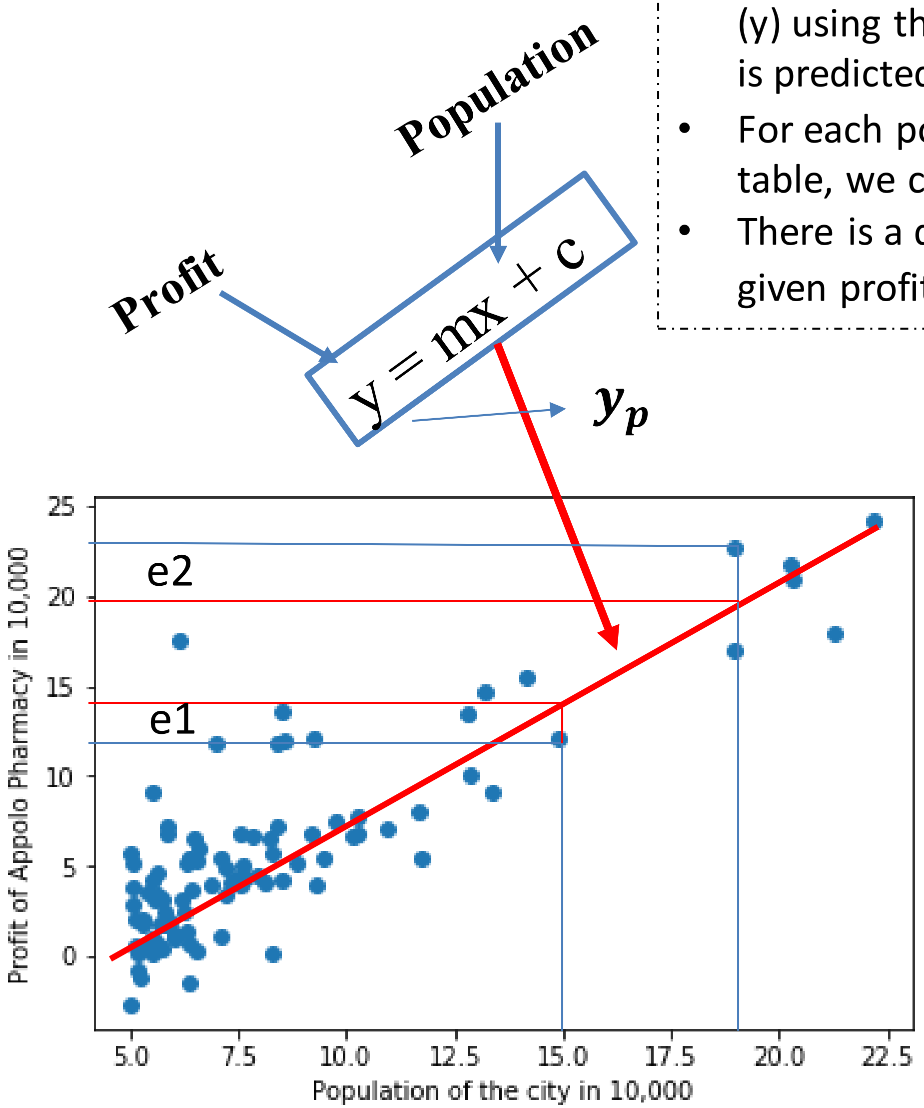
Single Variable Regression

Locations	Population of the city (in ten tho)	Profit ($ in ten thou)
0	6.1101	17.592
1	5.5277	9.1302
2	8.5186	13.662
3	7.0032	11.854
4	5.8598	6.8233
5	8.3829	11.886
6	7.4764	4.3483
7	8.5781	12
8	6.4862	6.5987
9	5.0546	3.8166
10	5.7107	3.2522
11	14.164	15.505
12	5.734	3.1551
13	8.4084	7.2258

Snippet of Data, there may be 13K or more rows.

Multiple lines can be fit to approximate linear relationship between Population and Profit. Say red one is the best fit line.





- For Every, x (Population) We can calculate a profit (y) using the approximated eqn, y=mx+c . The profit is predicted and we can call it → y_p.
- For each population there is already a profit given in table, we call it → y_o.
- There is a difference between predicted profit and given profit as = $(y_p - y_o)$,which is error 'e'

Locations	Population of the city (in ten tho)	Profit ($ in ten thou)
0	6.1101	17.592
1	5.5277	9.1302
2	8.5186	13.662
3	7.0032	11.854
4	5.8598	6.8233
5	8.3829	11.886
6	7.4764	4.3483
7	8.5781	12
8	6.4862	6.5987
9	5.0546	3.8166
10	5.7107	3.2522
11	14.164	15.505
12	5.734	3.1551
13	8.4084	7.2258

$e1 = (y_p - y_o)$

e2=

e3=

.

.

.

em

For 'm' data points

There will be 'm' error.

* Only e1 and e2 are shown in Image.

Mean of the sum of the squares of all the Errors (MSE).Here written as Cost (function).→

$$\text{Cost} = \frac{1}{2m}\sum_{i=1}^{m}(y_p - y_o)^2$$

The line with least Cost, is the best fit line. We have to find that.

Writing , y=mx+c in Matrix Form

Matrix Multiplication

$$\begin{bmatrix} 3 & 4 \\ 2 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 5 \\ 3 & 7 \end{bmatrix} = \begin{bmatrix} 3+12 & 15+28 \\ 2+3 & 10+7 \end{bmatrix}$$

Matrix 1 Matrix 2

$$= \begin{bmatrix} 15 & 43 \\ 5 & 17 \end{bmatrix}$$

Resultant Matrix

Dot product of

$\vec{a} = (a_1, a_2) = a_1 i + a_2 j$

$\vec{b} = (b_1, b_2) = b_1 i + b_2 j$

$\vec{a}.\vec{b} = a_1 . b_1 + a_2 . b_2$

Substituting in Cost Fn.

$$\text{Cost} = \frac{1}{2m} \sum_{i=1}^{m} (y_p - y_o)^2$$

$$= \frac{1}{2m} \sum_{i=1}^{m} (\theta_0 + \theta_1 x - y_o)^2$$

$y_p = mx + c$

Changing notation
$C \rightarrow \theta_0$ & $m \rightarrow \theta_1$

$or, y_p = \theta_1 x + \theta_0$

$or, y_p = \theta_0 + \theta_1 x$

or, $y_p = \theta_0 . 1 + \theta_1 . x$

$$y_p = \begin{bmatrix} 1 & x \end{bmatrix} * \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}$$

$\vec{x} = (1, x)$ *And* as a matrix [1 x]

$\vec{\theta} = (\theta_0, \theta_1)$ And as a matrix $[\theta_0 \theta_1]$

Vector dot product: $\vec{x} . \vec{\theta}$

Matrix product: $x\theta^T$

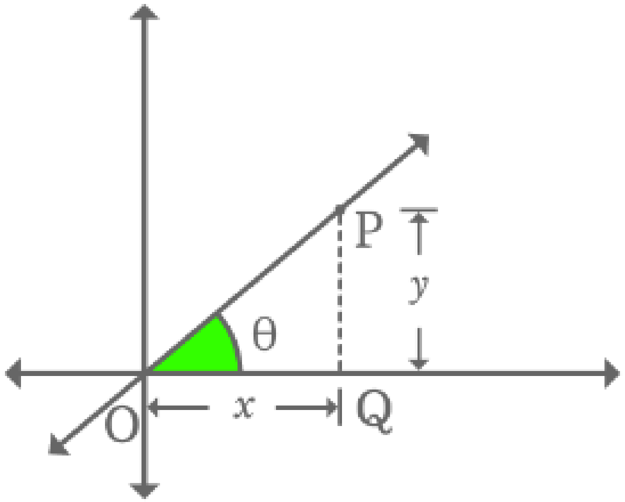
Ramendra Kumar

$$\text{Cost} = \frac{1}{2m}\sum_{i=1}^{m}(y_p - y_o)^2$$

$$= \frac{1}{2m}\sum_{i=1}^{m}(\theta_0 + \theta_1 x - y_o)^2$$

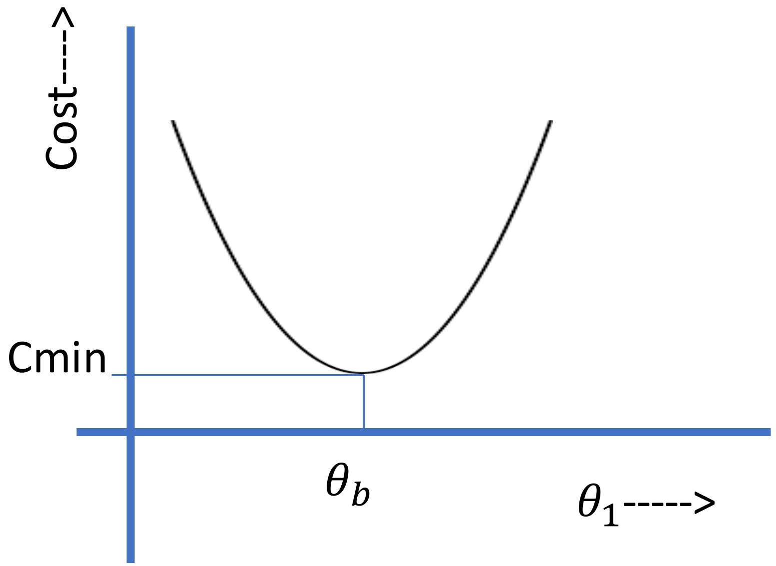
For time being, Assume approximated line passes through Origin, then y_p = mx + c → reduces to y_p = mx, according to changed notation, $y_p = \theta_1 x$

$$C = \frac{1}{2m}\sum_{i=1}^{m}(\theta_1 x - y_o)^2$$



Locations	Population of the city (in ten tho)	Profit ($ in ten thou)
0	6.1101	17.592
1	5.5277	9.1302
2	8.5186	13.662
3	7.0032	11.854
4	5.8598	6.8233
5	8.3829	11.886
6	7.4764	4.3483
7	8.5781	12
8	6.4862	6.5987
9	5.0546	3.8166
10	5.7107	3.2522
11	14.164	15.505
12	5.734	3.1551
13	8.4084	7.2258

x & y_o *both known (given data in table).*
So Cost 'C' is the function of θ_1 only and being Quadratic eqn, plot between C and θ_1 is parabolic as given below:

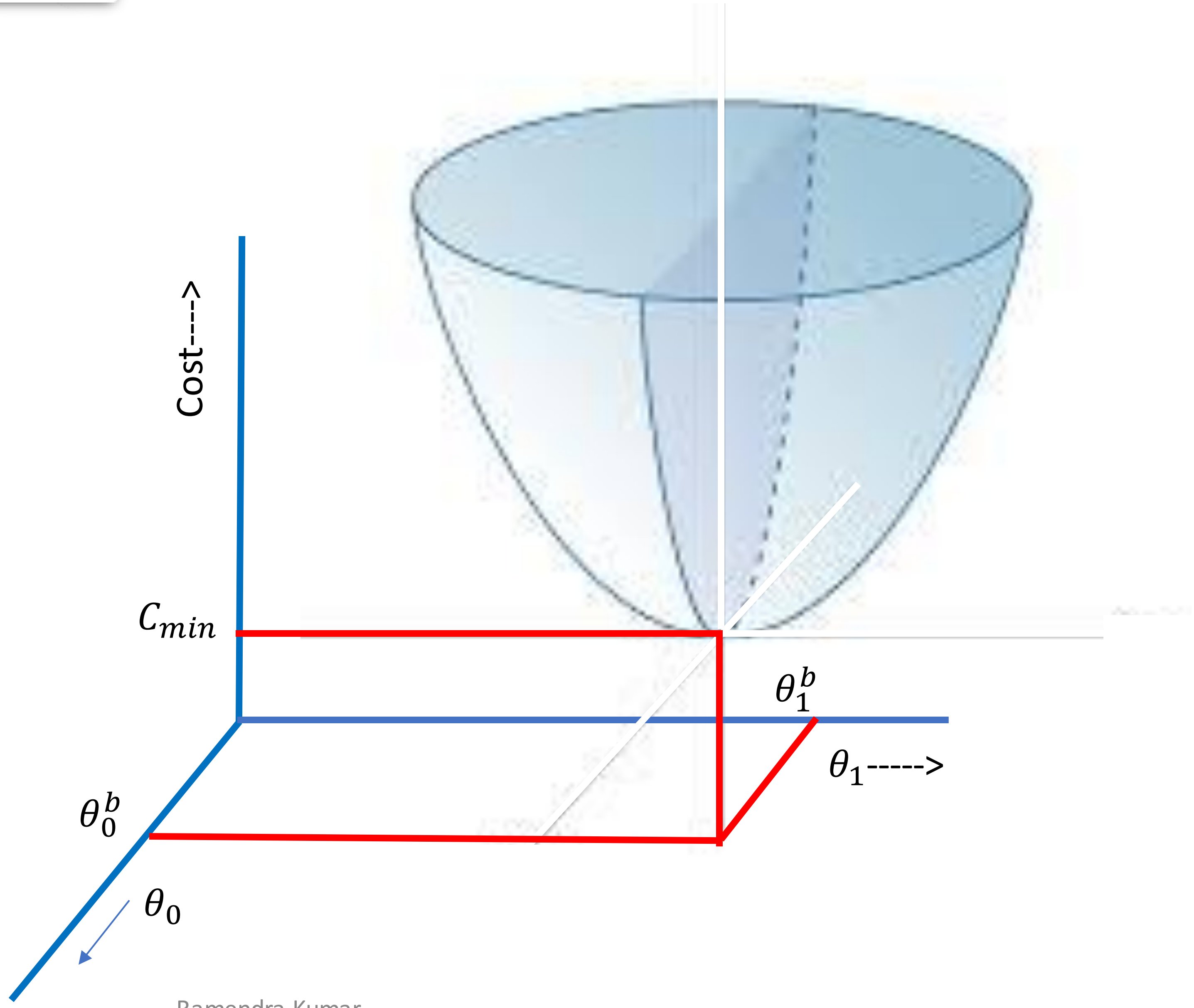


Plotting Cost Function, taking both θ_0 and θ_1

$$\text{Cost} = \frac{1}{2m}\sum_{i=1}^{m}(y_p - y_o)^2$$

$$= \frac{1}{2m}\sum_{i=1}^{m}(\theta_0 + \theta_1 x - y_o)^2$$

Equation of 3D Paraboloid



Basic Differentiation Formulas

$$\frac{dk}{dx} = 0 \qquad \text{where } k = \text{constant}$$

$$\frac{d(x)}{dx} = 1$$

$$\frac{d(kx)}{dx} = k \qquad \text{where } k = \text{constant}$$

$$\frac{d(x^n)}{dx} = nx^{n-1}$$

Partial Derivative Example

Given → x=1, y=2

$$z = 3x^2 + 2xy - y^2$$
$$= 3(1)^2 + 2(1)(2) - (2)^2$$
$$= 3$$

$$\frac{\partial z}{\partial x} = 6x + 2y = 6(1) + 2(2) = 10$$

$$\frac{\partial z}{\partial x} = 2x + 2y = 2(1) + 2(2) = 6$$

This should be differential with respect to y .
And following calculation is also wrong
try yourself.

$$\text{Cost} = \frac{1}{2m}\sum_{i=1}^{m}(y_p - y_o)^2$$

$$C = \frac{1}{2m}\sum_{i=1}^{m}(\theta_0 + \theta_1 x - y_o)^2$$

When $\theta_0 = 0$

$$C = \frac{1}{2m}\sum_{i=1}^{m}(\theta_1 x - y_o)^2$$

Defining Gradient and Calculating it.

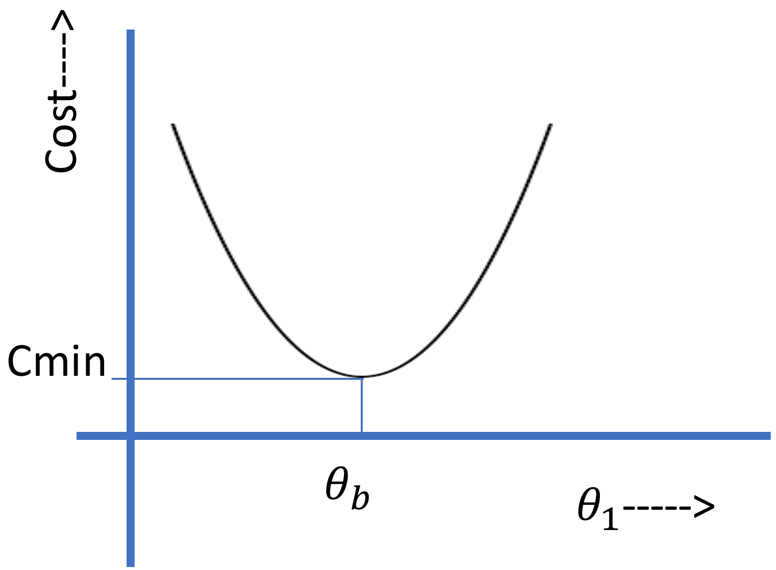
$$\mathbf{C} = \frac{1}{2m}\sum_{i=1}^{m}(\theta_1 x - y_o)^2$$

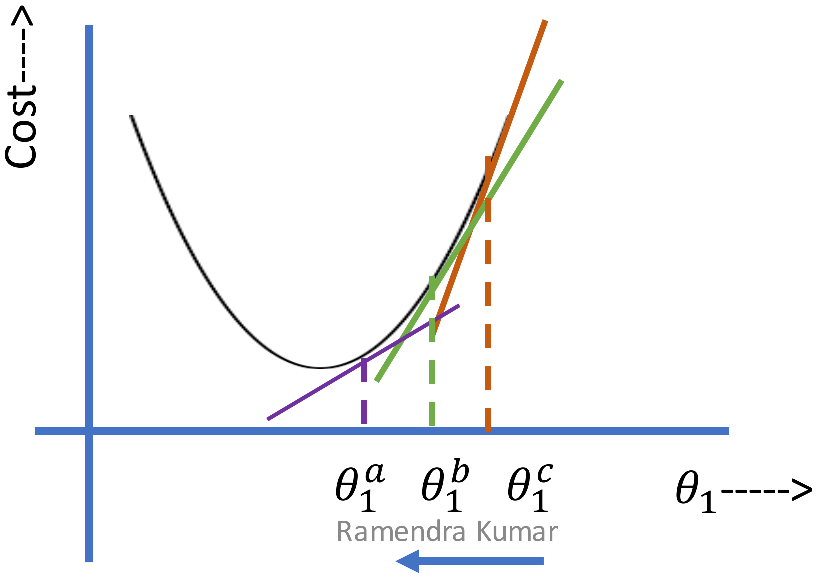
$$\frac{dC}{d\theta_1} = \frac{1}{2m}\sum_{i=1}^{m}\frac{d(\theta_1 x - y_o)^2}{d(\theta_1 x - y_o)} \cdot \frac{d(\theta_1 x - y_o)}{d(\theta_1)}$$

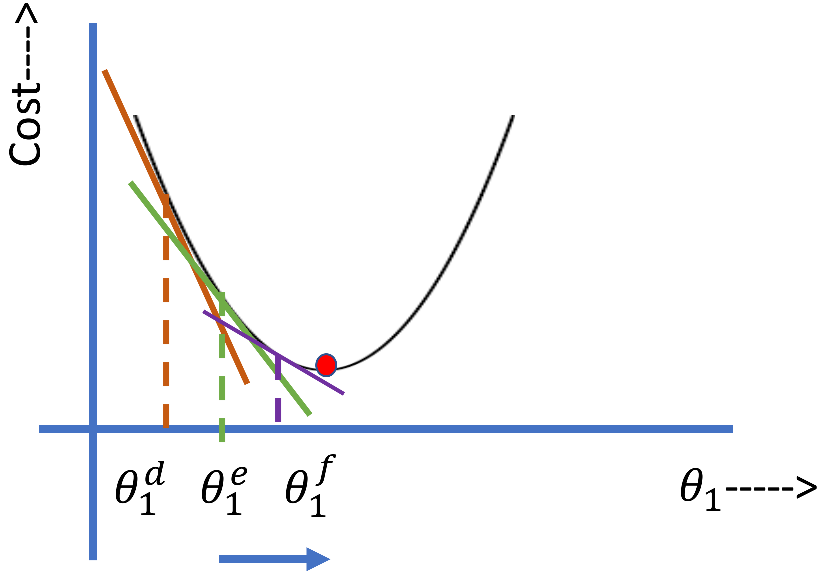
$$\frac{dC}{d\theta_1} = 2\,\frac{1}{2m}\sum_{i=1}^{m}(\theta_1 x - y_o).x$$

$$= \frac{1}{m}\sum_{i=1}^{m}(\theta_1 x - y_o).x$$

Gradient → Slope of Tangent at any given point θ_1 on the curve







Gradient Calculation of Full fledge Equation w.r.t θ_0 and θ_1

$$\text{Cost} = \frac{1}{2m}\sum_{i=1}^{m}(y_p - y_o)^2 \qquad = \frac{1}{2m}\sum_{i=1}^{m}(\theta_0 + \theta_1 x - y_o)^2$$

$$\frac{\partial C}{\partial \theta_0} = \frac{1}{2m}\sum_{i=1}^{m} \frac{\partial(\theta_0 + \theta_1 x - y_o)^2}{\partial(\theta_0 + \theta_1 x - y_o)} \cdot \frac{\partial(\theta_0 + \theta_1 x - y_o)}{\partial(\theta_0)}$$

$$\frac{\partial C}{\partial \theta_0} = 2\,\frac{1}{2m}\sum_{i=1}^{m}(\theta_0 + \theta_1 x - y_o)\,.\,1 \quad = \frac{1}{m}\sum_{i=1}^{m}(\theta_0 + \theta_1 x - y_o)$$

$$\boxed{\frac{\partial C}{\partial \theta_0} = \frac{1}{m}\sum_{i=1}^{m}(\theta_0 + \theta_1 x - y_o)}$$

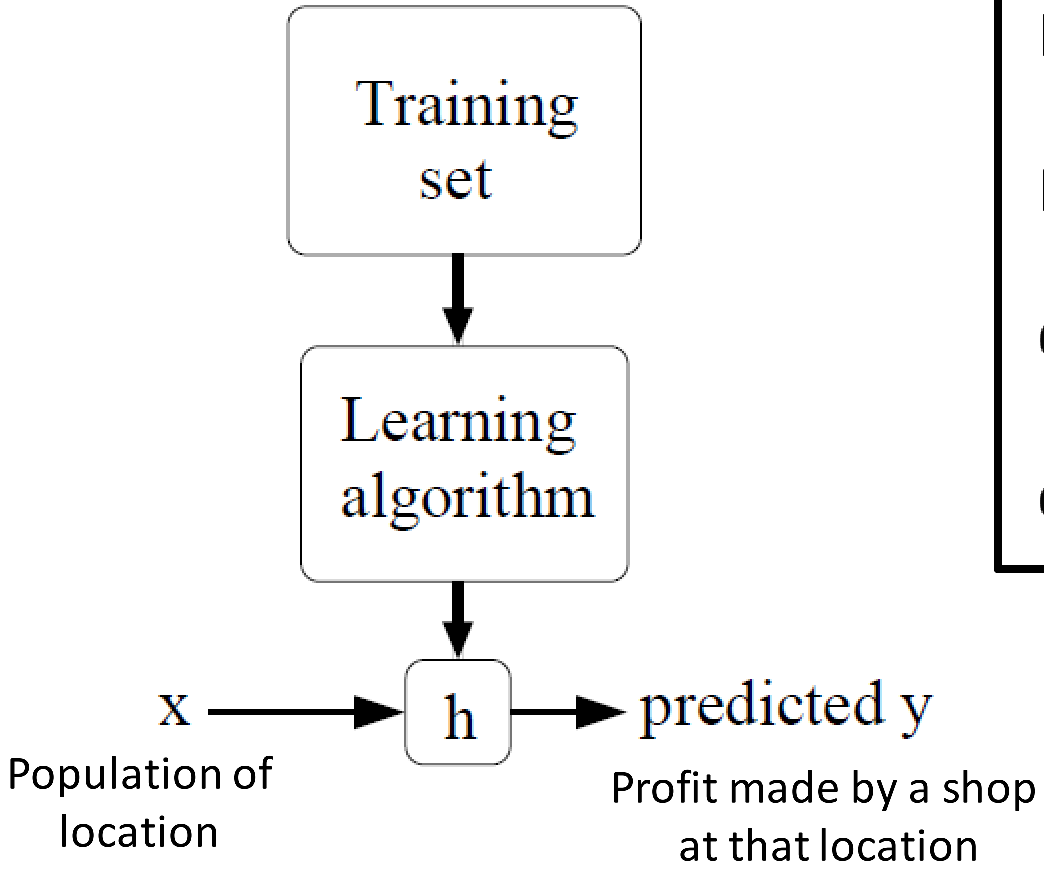
$$\frac{\partial C}{\partial \theta_1} = \frac{1}{2m}\sum_{i=1}^{m} \frac{\partial(\theta_0 + \theta_1 x - y_o)^2}{\partial(\theta_0 + \theta_1 x - y_o)} \cdot \frac{\partial(\theta_0 + \theta_1 x - y_o)}{\partial(\theta_1)}$$

$$\frac{\partial C}{\partial \theta_1} = 2\,\frac{1}{2m}\sum_{i=1}^{m}(\theta_0 + \theta_1 x - y_o)\,.\,x \quad = \frac{1}{m}\sum_{i=1}^{m}(\theta_0 + \theta_1 x - y_o).\,x$$

$$\boxed{\frac{\partial C}{\partial \theta_1} = \frac{1}{m}\sum_{i=1}^{m}(\theta_0 + \theta_1 x - y_o).\,x}$$

Formal Definition

Our goal is, given a training set, to learn a function h : X → Y so that h(x) is a “good” predictor for the corresponding value of y. For historical reasons, this function h is called a hypothesis.



Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$ $\quad$ $y_p = mx + c$

Parameters: θ_0, θ_1

Cost Function: $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

$$\text{Cost} = \frac{1}{2m} \sum_{i=1}^{m} (y_p - y_o)^2$$

$$= \frac{1}{2m} \sum_{i=1}^{m} (\theta_0 + \theta_1 x - y_o)^2$$

Understanding Gradient Descent

$$\text{Cost} = \frac{1}{2m}\sum_{i=1}^{m}(y_p - y_o)^2$$

$$= \frac{1}{2m}\sum_{i=1}^{m}(\theta_0 + \theta_1 x - y_o)^2$$

$$\boxed{\theta_1 := \theta_1 - \alpha \frac{dJ(\theta_1)}{d\theta_1}}$$

α=Learning Rate

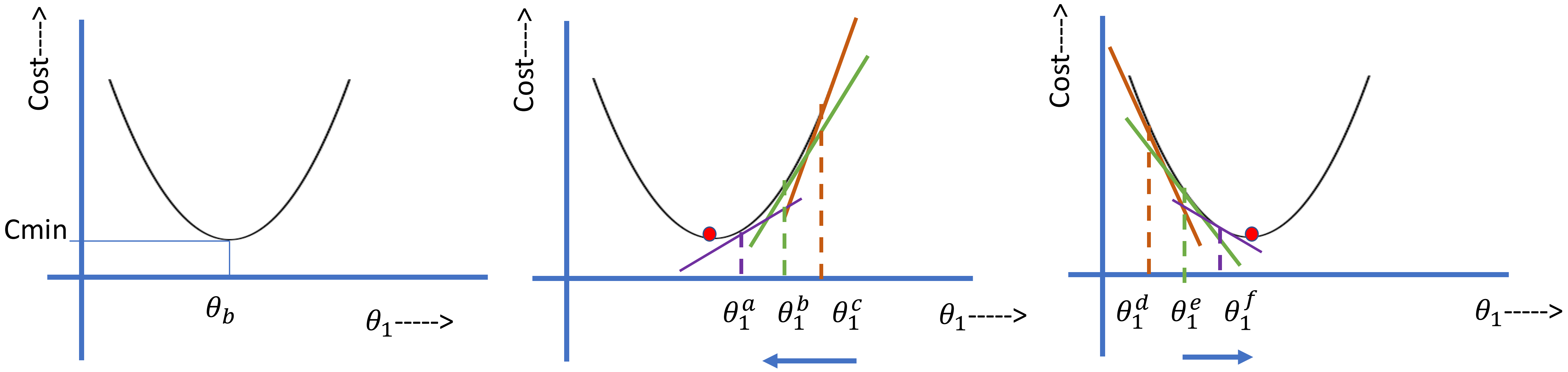
When $\theta_0 = 0$

$$\mathbf{C} = \frac{1}{2m}\sum_{i=1}^{m}(\theta_1 x - y_o)^2$$

$$\frac{dC}{d\theta_1} = \frac{1}{2m}\sum_{i=1}^{m}\frac{d(\theta_1 x - y_o)^2}{d(\theta_1 x - y_o)} \cdot \frac{d(\theta_1 x - y_o)}{d(\theta_1)}$$

$$\frac{dC}{d\theta_1} = 2\,\frac{1}{2m}\sum_{i=1}^{m}(\theta_1 x - y_o).\,x$$

$$= \frac{1}{m}\sum_{i=1}^{m}(\theta_1 x - y_o).\,x$$



Ramendra Kumar

Full Fledge Gradient Descent Algorithm

Have some function $J(\theta_0, \theta_1)$

Want $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$

Outline:

- Start with some θ_0, θ_1
- Keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$ until we hopefully end up at a minimum

$$\theta_0 := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$

repeat until convergence {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \quad (\text{for } j = 0 \text{ and } j = 1)$$

}

Error Calculation

1. **Mean Absolute Error** (MAE) is the mean of the absolute value of the errors. It is calculated as:

$$\text{MAE} = \frac{1}{n}\sum_{j=1}^{n}|y_j - y_j|$$

Mean Absolute Error

2. **Mean Squared Error** (MSE) is the mean of the squared errors and is calculated as:

$$MSE = \frac{1}{N}\sum_{i}^{n}(Y_i - y_i)^2$$

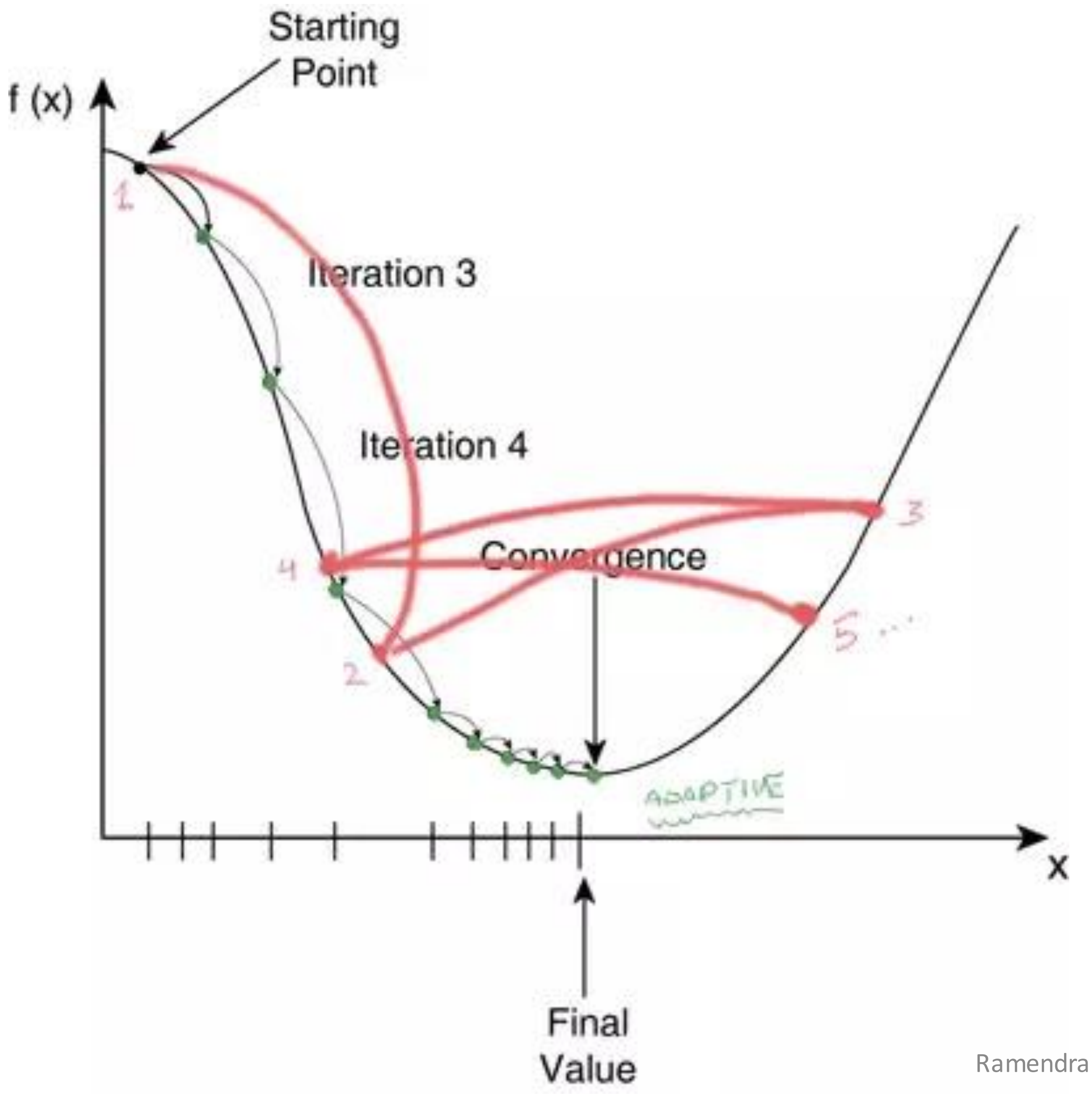
Mean Squared Error

3. **Root Mean Squared Error** (RMSE) is the square root of the mean of the squared errors:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

Root Mean Squared Error

Concept of Learning Rate



$$\theta_1 := \theta_1 - \alpha \frac{dJ(\theta_1)}{d\theta_1}$$

First Try , α = 0.0001

- 0.0001 X 3=0.0003 (2nd Try)
- 0.0003 X 3 ~ 0.001 (3rd Try, & so on)
- 0.001 X 3= 0.003
- 0.003 X 3 ~ 0.01

Multivariable Regression

$$Yp = M_1 X_1 + M_2 X_2 + \ldots + M_n X_n + C$$

$$Yp = \theta_0.1 + \theta_1 X_1 + \theta_2 X_2 + \theta_3 X_3 \ldots + \theta_n X_n$$

Up to n feature

$$y = \begin{bmatrix} 1 & X_1 & X_2 & X_3 & \ldots & \ldots & \ldots & X_n \\ 1 & X_1 & X_2 & X_3 & \ldots & \ldots & \ldots & X_n \\ 1 & X_1 & X_2 & X_3 & \ldots & \ldots & \ldots & X_n \end{bmatrix} * \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ . \\ . \\ . \\ \theta_n \end{bmatrix}$$

....

Up to m dataset

M X (n+1) **(n+1,1)**

```
#np.random.seed(0)
theta=np.random.randn(2,1)
```

(n+1,1)

...

```
n_iterations=10000
alpha=0.01
for iteration in range(n_iterations):
    grad=(1/m)*X.T.dot((X.dot(theta)-Y))
    theta=theta-alpha*grad ## Gradient descent
print(theta)
```

Same Grad.D. will work

$$Yp = mx + c \quad \text{or, } Yp = \theta_1 x + \theta_0$$

$$\text{or, } Yp = \theta_0 + \theta_1 x \quad \text{or, } Yp = \theta_0.1 + \theta_1.x$$

$$Yp = \begin{bmatrix} 1 & x \end{bmatrix} * \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}$$

Locations	Size of Flat(feet2)	Number of Bedrooms	Price ($)
0	2104	3	399900
1	1600	3	329900
2	2400	3	369000
3	1416	2	232000
4	3000	4	539900
5	1985	4	299900
6	1534	3	314900
7	1427	3	198999
8	1380	3	212000
9	1494	3	242500
10	1940	4	239999
11	2000	3	347000
12	1890	3	329999

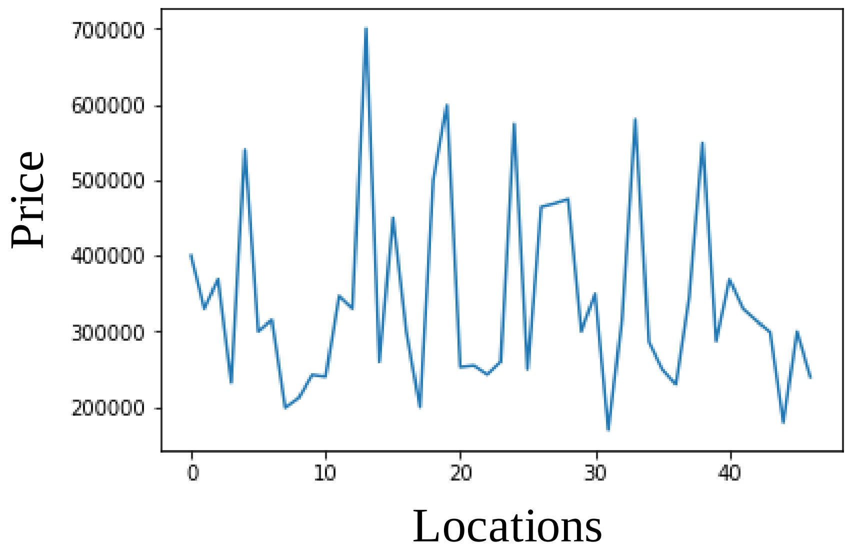
Price **Size of Flat** **Number of Bedrooms**

$$Yp = M_1 X_1 + M_2 X_2 + C$$

$$Yp = M_1 X_1 + M_2 X_2 + \ldots + M_n X_n + C$$

PREDICTION > MULTIVARIABLE REGRESSION

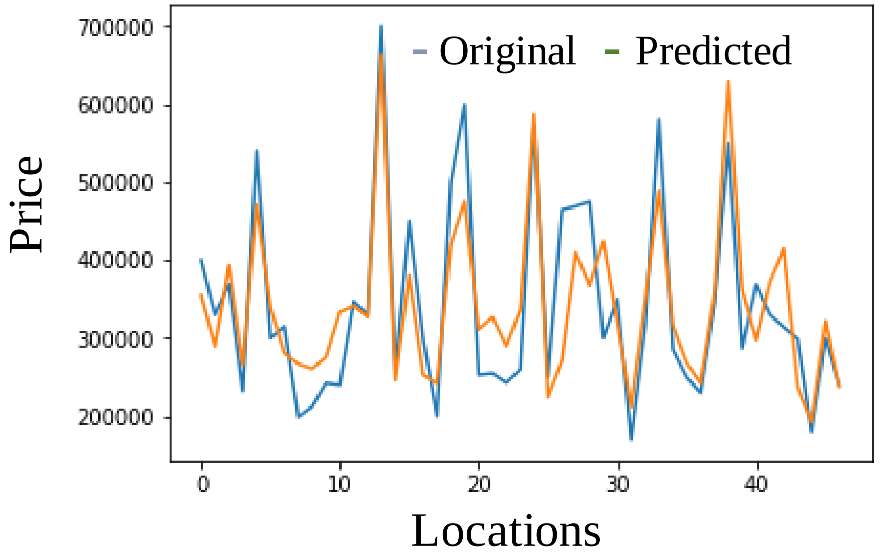
Locations	Size of Flat(feet2)	Number of Bedrooms	Price ($)
0	2104	3	399900
1	1600	3	329900
2	2400	3	369000
3	1416	2	232000
4	3000	4	539900
5	1985	4	299900
6	1534	3	314900
7	1427	3	198999
8	1380	3	212000
9	1494	3	242500
10	1940	4	239999
11	2000	3	347000
12	1890	3	329999



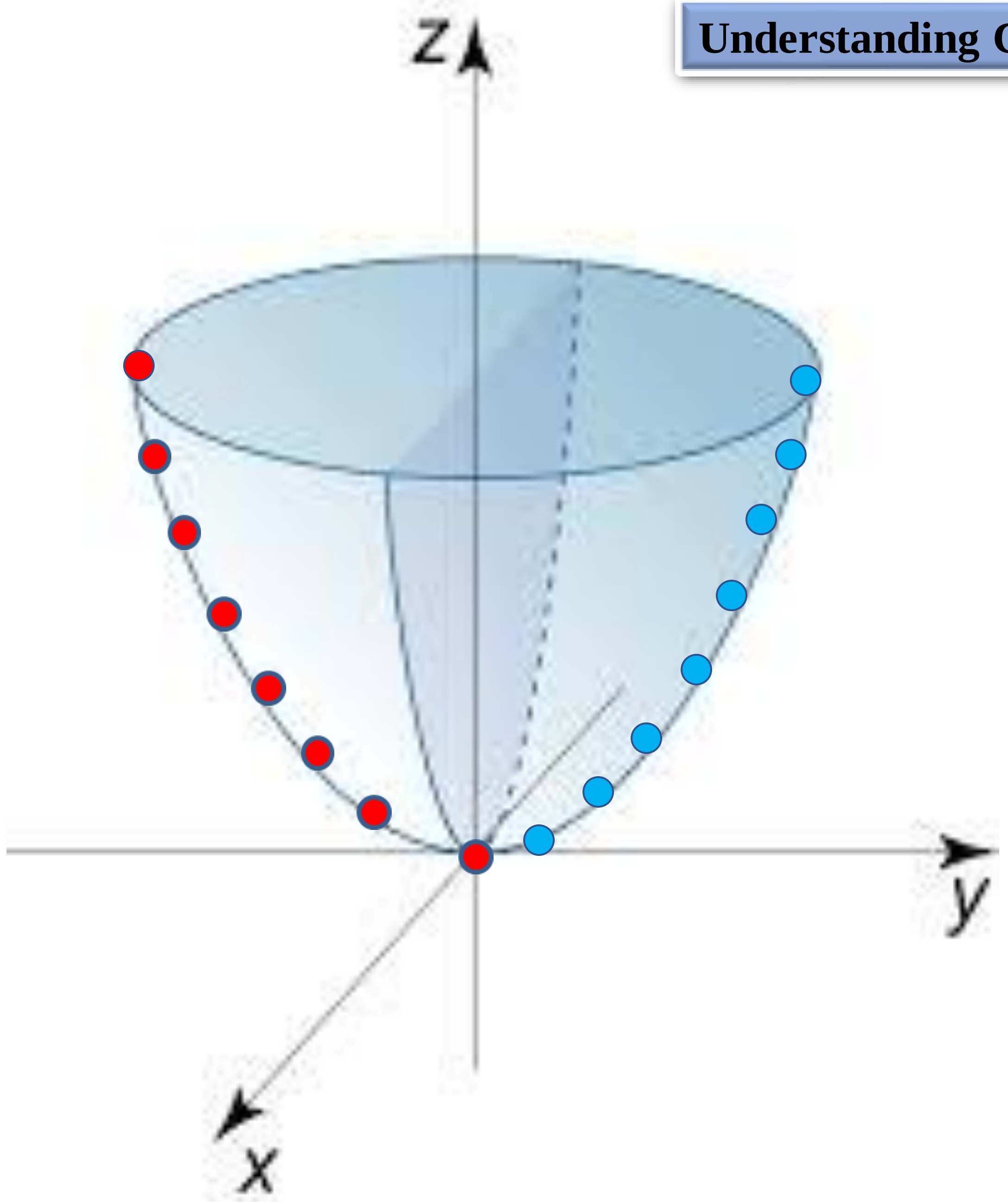
Price **Size of Flat** **Number of Bedrooms**

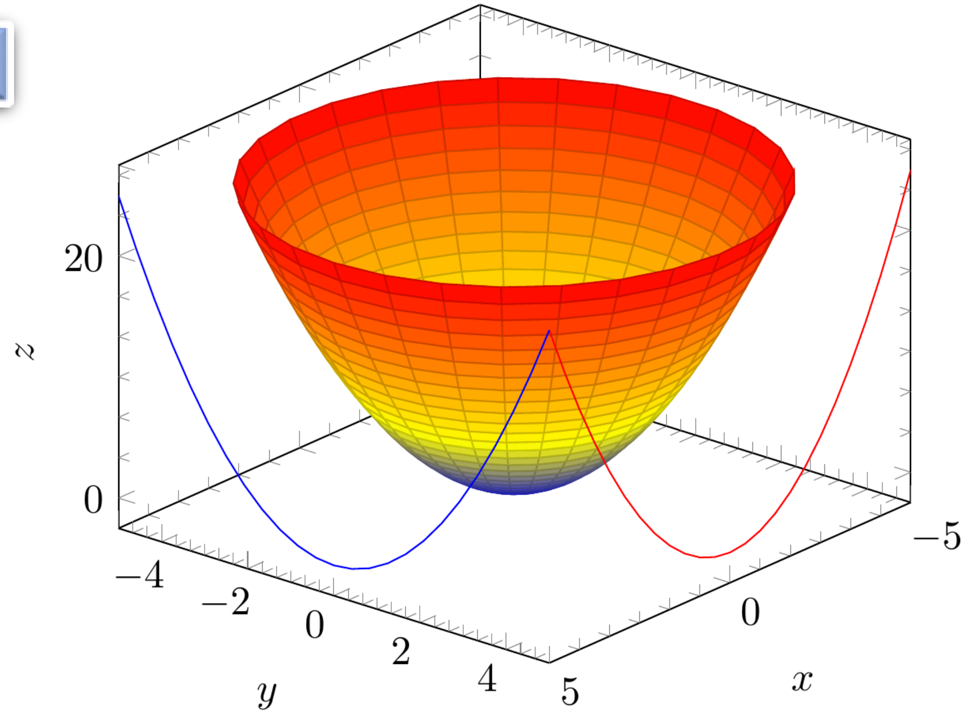
$$Yp = M_1 X_1 + M_2 X_2 + C$$

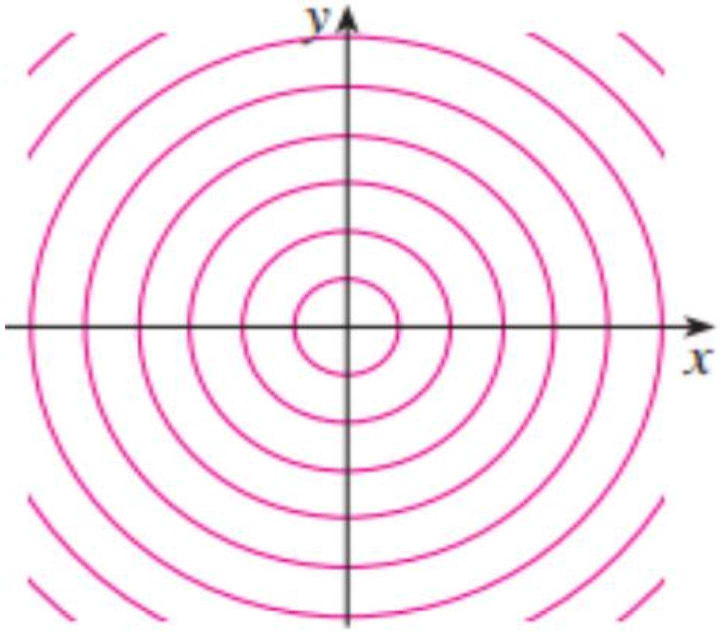
$$Yp = M_1 X_1 + M_2 X_2 + \ldots + M_n X_n + C$$



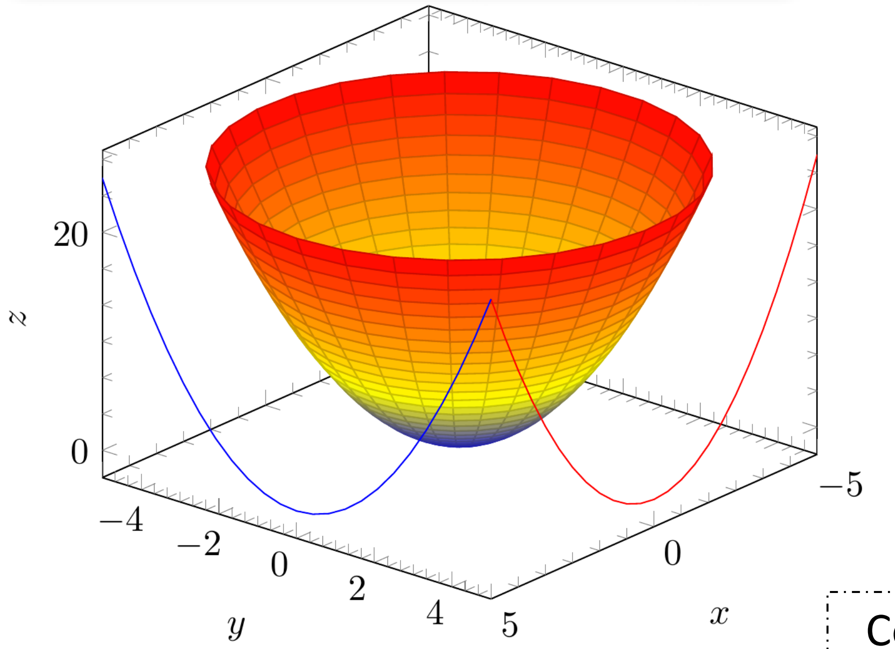
Understanding Contour Plot

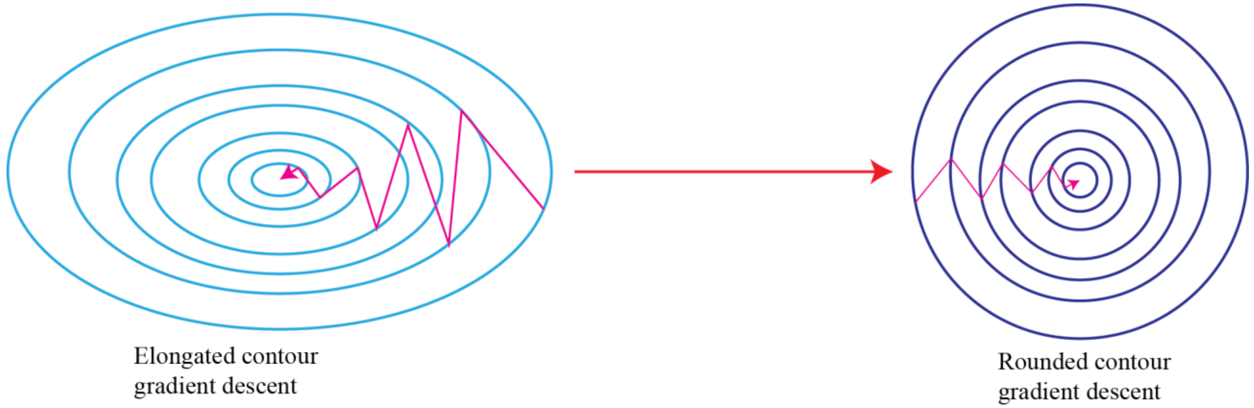


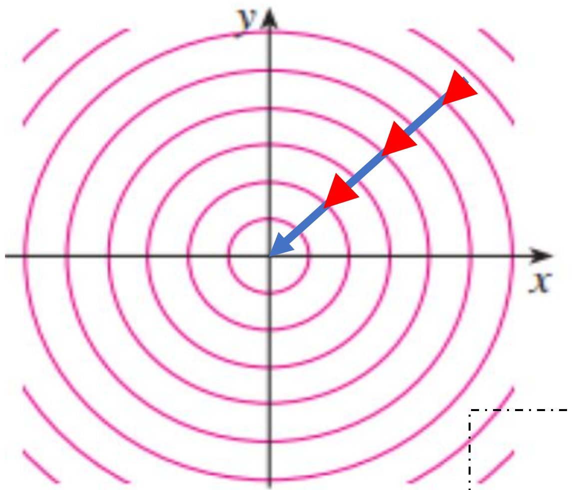




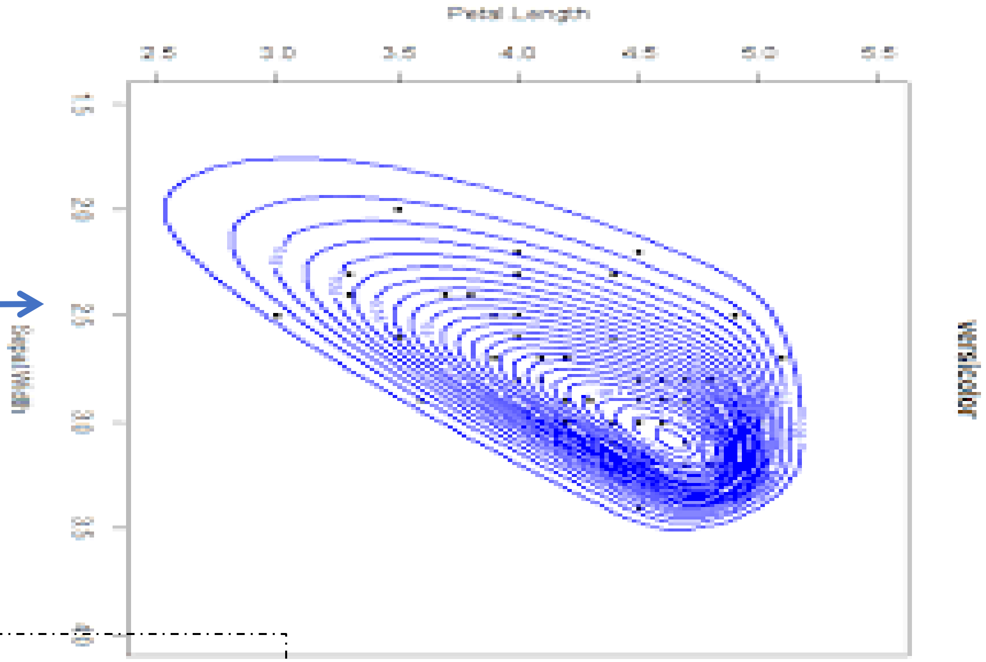
Skewed Data-Set and Standardization







Contour plot of skewed data
Cost fn.
Gradient Descent has tough time to reach at minima.
It may take too long
or, it may diverge completely.



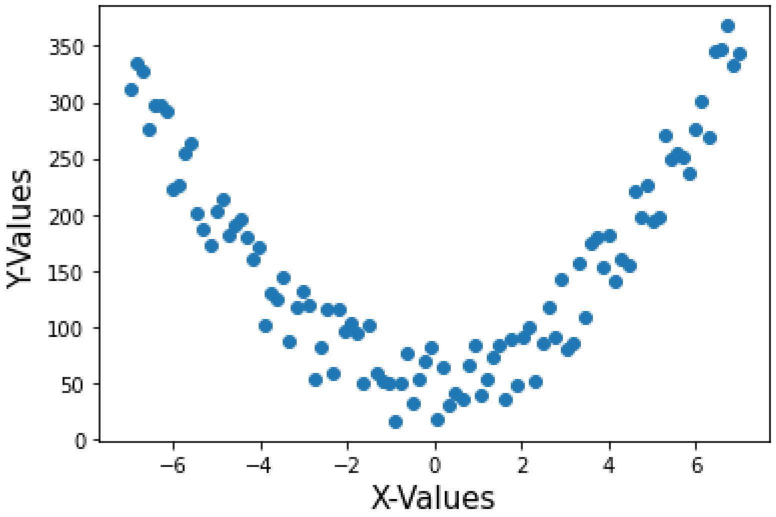
Standardization:
Each feature is transformed by Subtracting
its own Mean and Dividing by its own Standard Deviation.

Polynomial Regression

Third Degree Polynomial Feature Transformation

Second Degree Polynomial Feature Transformation

X	Y
-7	311
-6.85859	335.524
-6.71717	328.288
-6.57576	275.292
-6.43434	296.536
-6.29293	297.0199
-6.15152	292.7438
-6.0101	222.7077
-5.86869	225.9115
-5.72727	254.3554
-5.58586	264.0392
-5.44444	201.963
-5.30303	186.1267
-5.16162	173.5305
-5.0202	203.1742
-4.87879	214.0579
-4.73737	182.1815
-4.59596	190.5451
-4.45455	196.1488
-4.31313	179.9923
-4.17172	160.0759
-4.0303	170.3994
-3.88889	101.963



From the plot, it is seen that there is no linear relationship between X and Y.

$(1+X)^2=1+2x+X^2 = (1,X,X^2)$			
f1	f2	f3	Y
1	-7	49	311
1	-6.85859	47.0402	335.524
1	-6.71717	45.1204	328.288
1	-6.57576	43.24059	275.292
1	-6.43434	41.40078	296.536
1	-6.29293	39.60096	297.0199
1	-6.15152	37.84114	292.7438
1	-6.0101	36.12131	222.7077
1	-5.86869	34.44149	225.9115
1	-5.72727	32.80165	254.3554
1	-5.58586	31.20182	264.0392
1	-5.44444	29.64198	201.963
1	-5.30303	28.12213	186.1267
1	-5.16162	26.64228	173.5305
1	-5.0202	25.20243	203.1742
1	-4.87879	23.80257	214.0579
1	-4.73737	22.44271	182.1815
1	-4.59596	21.12284	190.5451
1	-4.45455	19.84298	196.1488

$(1+X)^3=1+3X+3X^2+X^3 = (1, X, X^2, X^3)$				
f1	f2	f3	f4	Y
1	-7	49	-343	311
1	-6.85859	47.0402	-322.629	335.524
1	-6.71717	45.1204	-303.081	328.288
1	-6.57576	43.24059	-284.34	275.292
1	-6.43434	41.40078	-266.387	296.536
1	-6.29293	39.60096	-249.206	297.0199
1	-6.15152	37.84114	-232.78	292.7438
1	-6.0101	36.12131	-217.093	222.7077
1	-5.86869	34.44149	-202.126	225.9115
1	-5.72727	32.80165	-187.864	254.3554
1	-5.58586	31.20182	-174.289	264.0392
1	-5.44444	29.64198	-161.384	201.963
1	-5.30303	28.12213	-149.133	186.1267
1	-5.16162	26.64228	-137.517	173.5305
1	-5.0202	25.20243	-126.521	203.1742
1	-4.87879	23.80257	-116.128	214.0579
1	-4.73737	22.44271	-106.32	182.1815
1	-4.59596	21.12284	-97.0797	190.5451
1	-4.45455	19.84298	-88.3914	196.1488

Now, instead of using ‘X ‘ as input features, We will use f1,f2,f3,… as features for modeling . So it becomes simply a multivariable regression problem.

THANK YOU !!

Reference: http://cs229.stanford.edu/

www.linkedin.com/in/ramendra-kumar-57334478