**Introduction**

• What is Machine Learning?

# Machine Learning: A Definition

**Definition:** The field of study that gives computers the ability to learn without being explicitly learned.

(Arthur Samuel-1950)

# Machine Learning: A Definition

**Definition:** A computerprogram is said to *learn* from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

(Tom Mitchell-1998)

## Case study 1: Spam/Not spam emails

• Suppose your email program watches which emails you do or do not mark as spam, and based on that learns how to better filter spam. What is the task T in this setting?

* **The task T** is classifying emails as spam or not spam
* **The experience E** is watching you label emails as spam or not spam
* **The performance P** is the number of emails correctly classified as spam or not spam

## Case study 2: Handwriting recognition learning

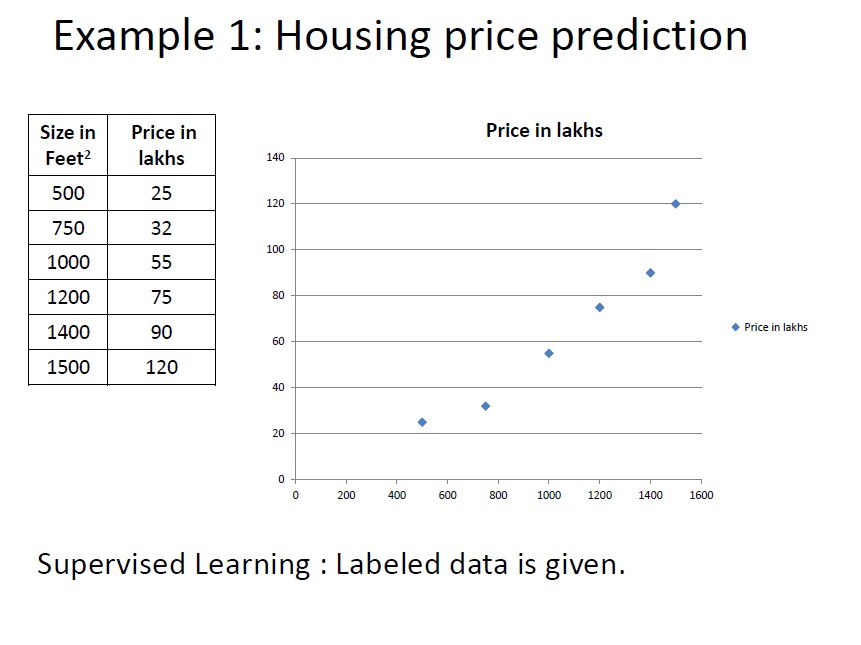
* **Task T :** recognizing and classifying handwritten words within images
* **Performance measure P :** percent of words correctly classified
* **Training experience E:** a database of handwritten words with given classifications

# Machine Learning Approaches

* Supervised learning
* Un-supervised learning

# Supervised Learning

* Input and output labels are given.
* Categorized into:
  + regression and
  + Classification
* Regression:
  + map input variables to some continuous function
  + predict results within a continuous output
* Classification:
  + map input variables into discrete categories
  + predict results in a discrete output



Size in Feet

2

P

r

i

c

e

i

n

l

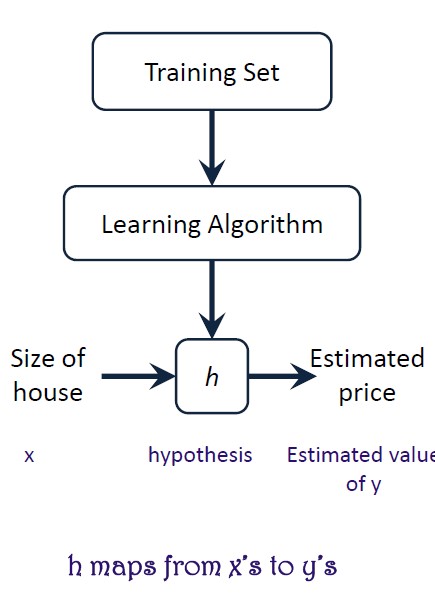
a

k

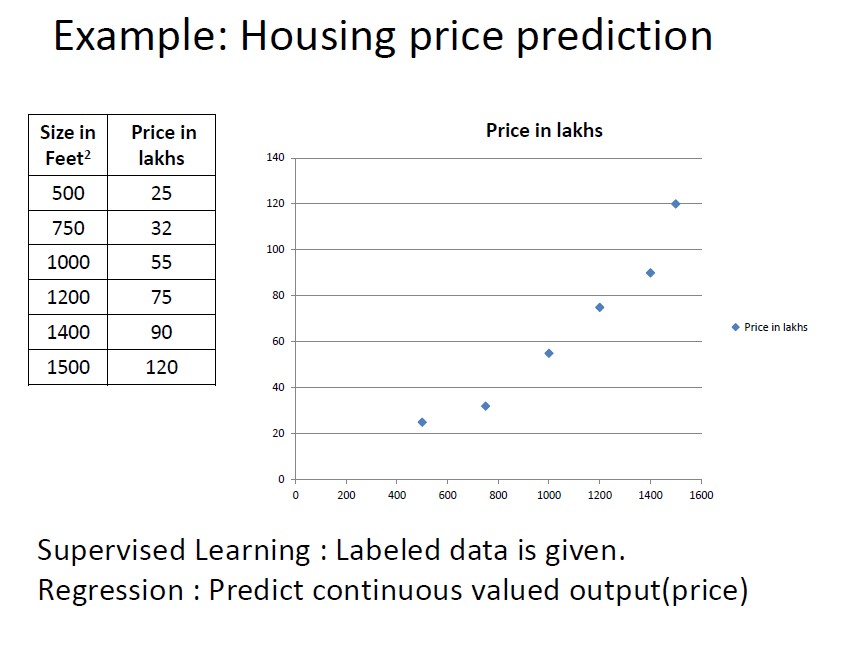
h

s

**Housing Price Prediction**



**10**



Size in Feet

2

P

r

i

c

e

i

n

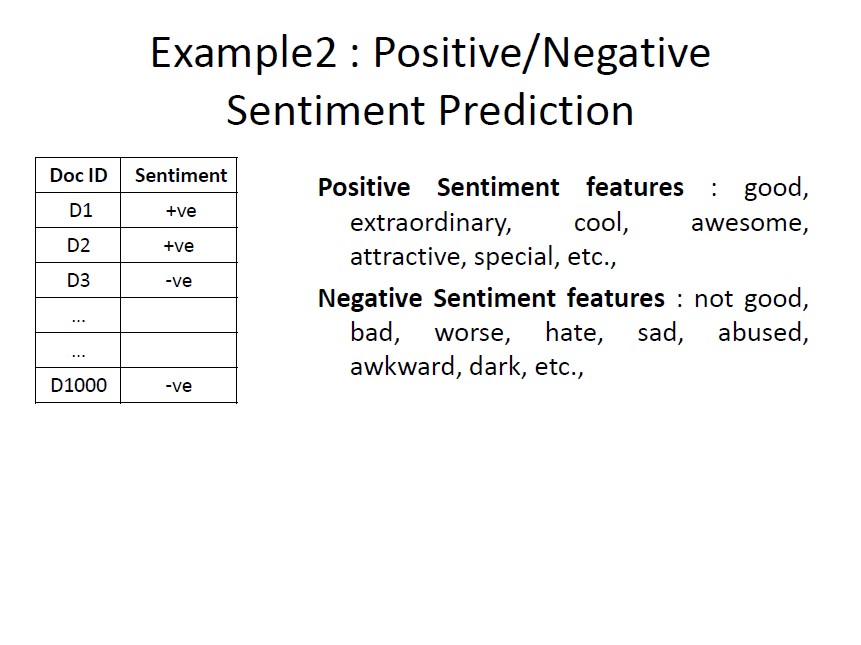
l

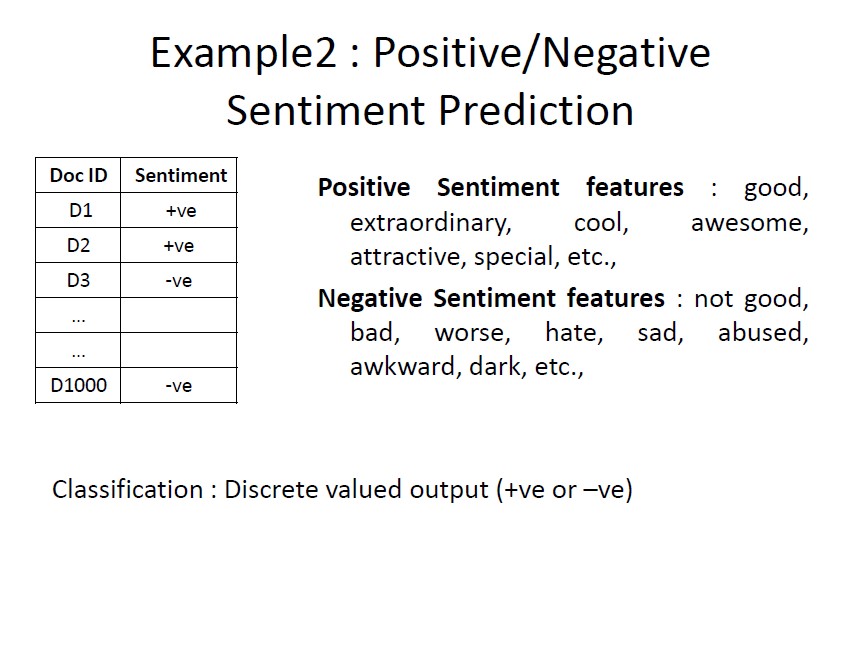
a

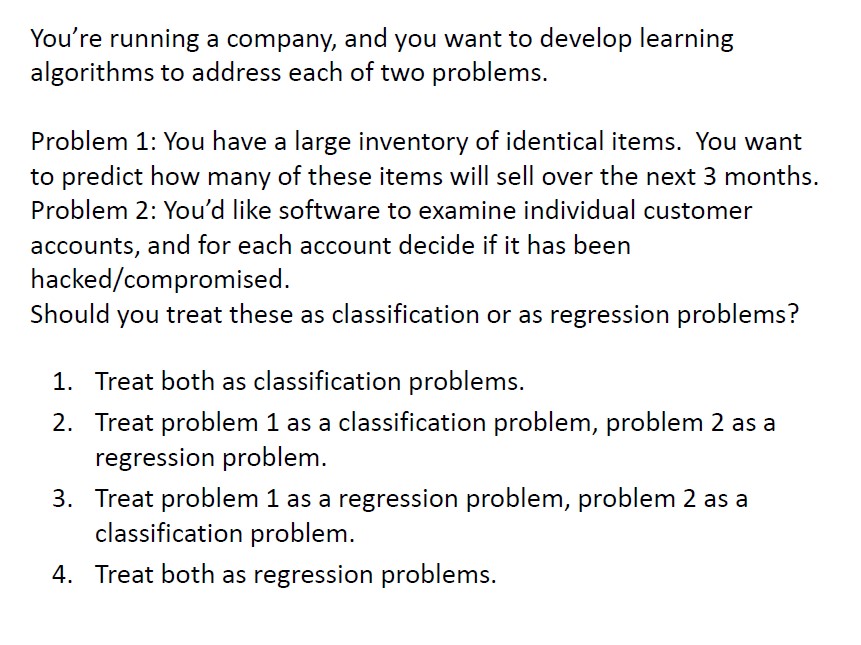
k

h

s







### Answer

Treat problem 1 as a regression problem, problem 2 as a classification problem.

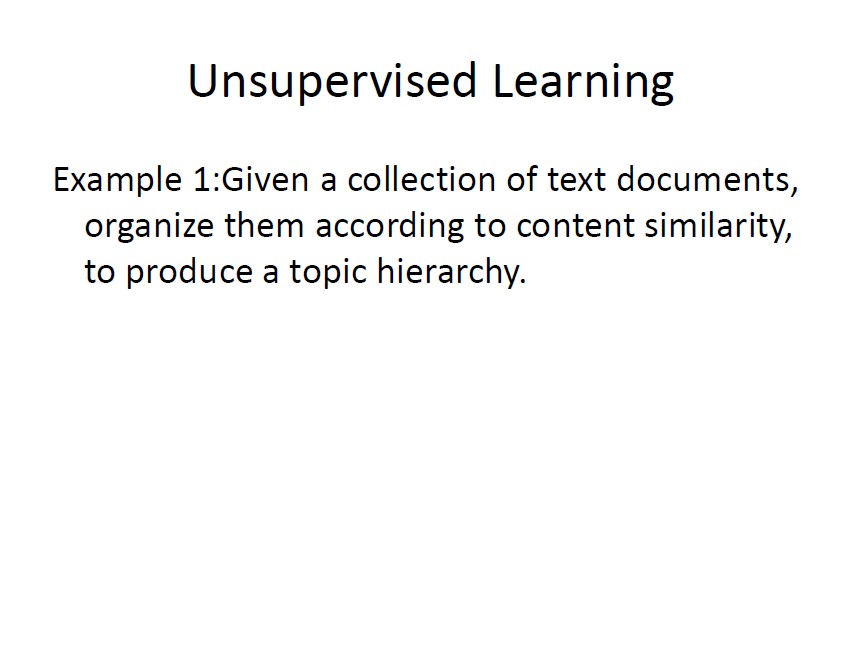
### Explanation

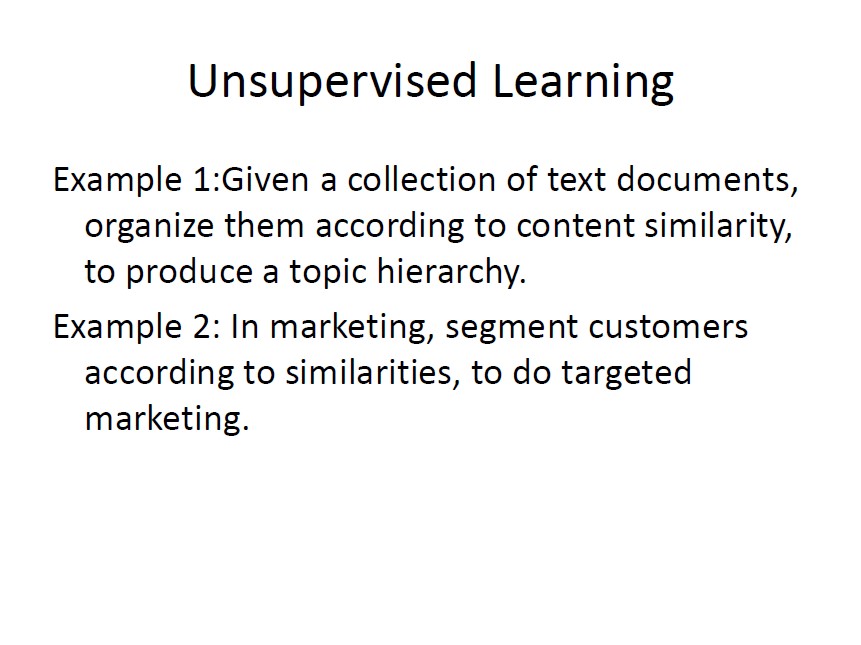
Problem 1 has Continuous Valued Output.  
Problem 2 has Discrete Valued Output.

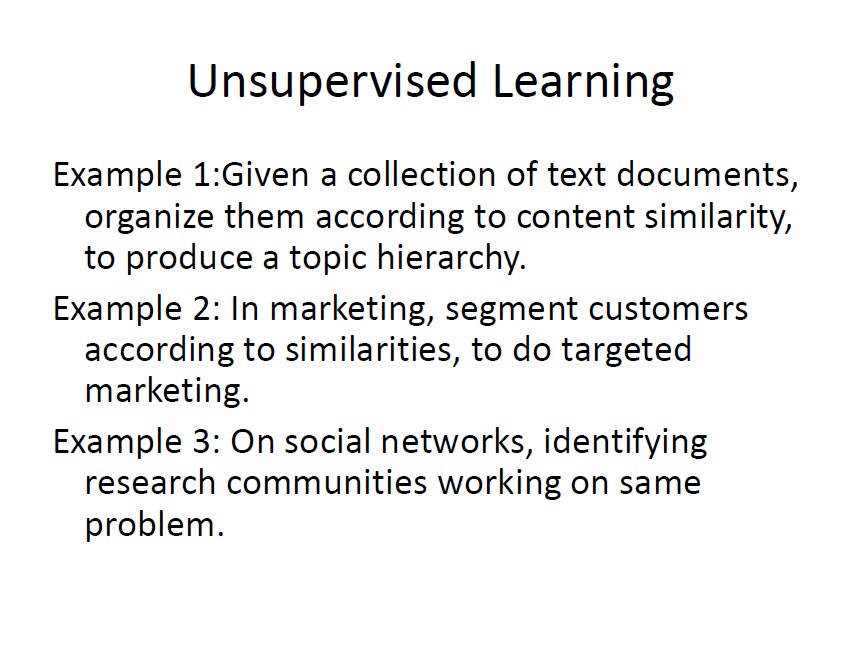
**Unsupervised Learning**

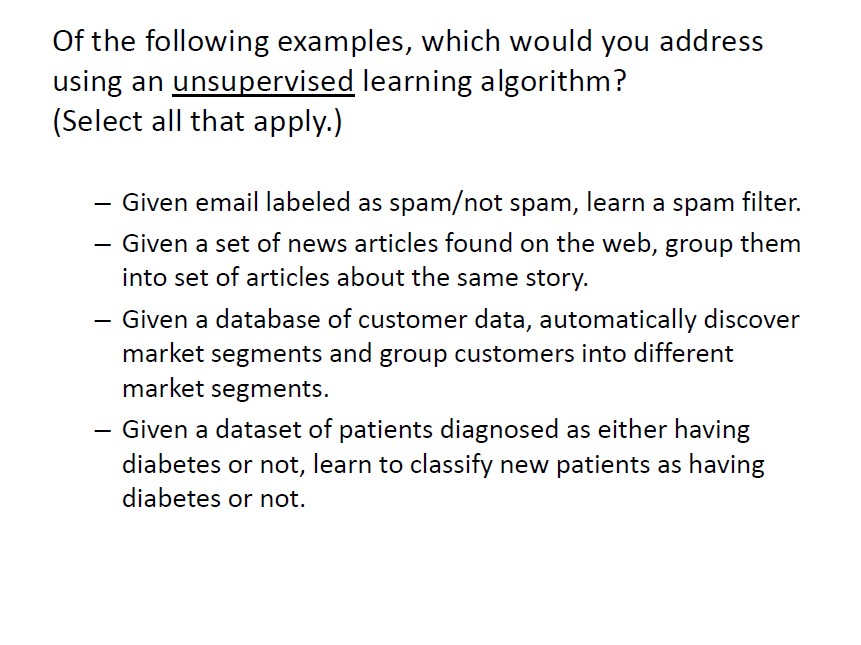
# Unsupervised Learning

• Input is known but output is not known.









# Ans 1.S,2.US,3.US,

# Linear Regression

## Linear Regression with one variable

* **Single feature(variable)**

|  |  |
| --- | --- |
| **Size (feet2)** | **Price (lakhs)** |
| 2104 | 460 |
| 1416 | 232 |
| 1534 | 315 |
| 852 | 178 |
| … | … |

* **Model**

hw(x) = w0 + w1\*x1

## Linear Regression with one variable

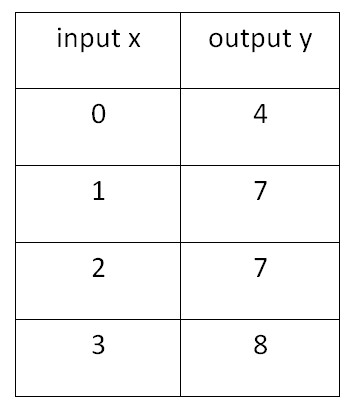
* **Single feature(variable)**

|  |  |
| --- | --- |
| **Size (feet2) x** | **Price (lakhs) y** |
| 2104 | 460 |
| 1416 | 232 |
| 1534 | 315 |
| 852 | 178 |
| … | … |

* **Hypothesis Model**  hw (x) = w0 + w1\*x

Example:

* Suppose we have the following set of training data:



* Now we can make a random guess about our hw(x) function: w0​=2 and w1​=2. The hypothesis function becomes hw(x)=2+2x.
* For input of 1 , hypothesis, y will be 4

## Linear Regression with Multiple variables

* **Multiple features(variables)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Size (feet2)**    **x1** | **Number of bedrooms x2** | **Number of floors x3** | **Age of home**  **(years) x4** | **Price (lakhs)**    **Y** |
| 2104 | 5 | 1 | 45 | 460 |
| 1416 | 3 | 2 | 40 | 232 |
| 1534 | 3 | 2 | 30 | 315 |
| 852 | 2 | 1 | 36 | 178 |
| … | … | … | … | … |

* **Hypothesis Model**

hw (x) = h(x)= w0 + w1\*x1+ w2\*x2 + w3\*x3 +w4\*x4

## Hypothesis

h(x) = w0 + w1\*x1+ w2\*x2 + w3\*x3 +w4\*x4 Example:

h(x) = 80 + 0.1\*x1+ 0.01 \*x2 + 3\*x3 – 2 \*w4

Now,

**Input is a vector of the form** **W’s is a vector of the form**

x0 w0

X = x1∈ Rn+1 W= w1 ∈ Rn+1 x2 w2 x3 w3

… … xn wn

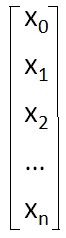
## Hypothesis

**Input is a vector of the form** **W’s is a vector of the form**

x0 w0

X = x1 ∈ Rn+1 W= w1 ∈ Rn+1

…x2 w… 2

 xn wn

Now, h(x) = w0 + w1\*x1+ w2\*x2 + w3\*x3 +w4\*x4

For convenience of notation, define x0=1

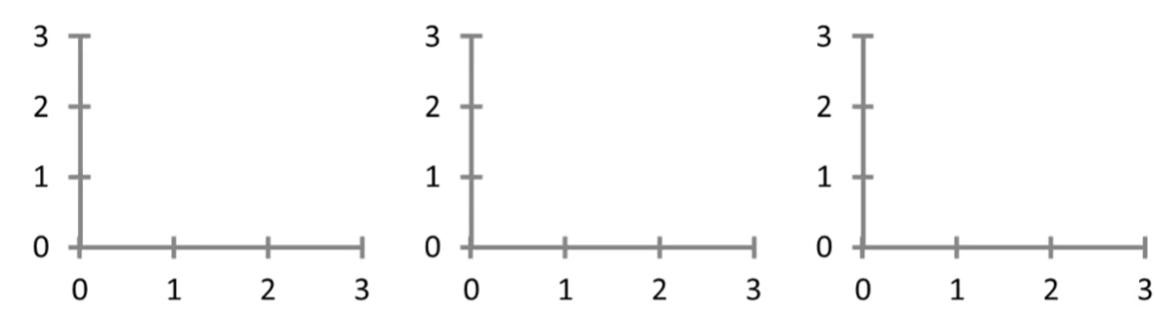
Therefore, h(x) = WTX = [w0 w1 w2 … wn]

(n+1) x 1 matrix 1x (n+1)  matrix

# Cost Function

## Hypothesis function: h (x) = w + w \*x

Price



Price

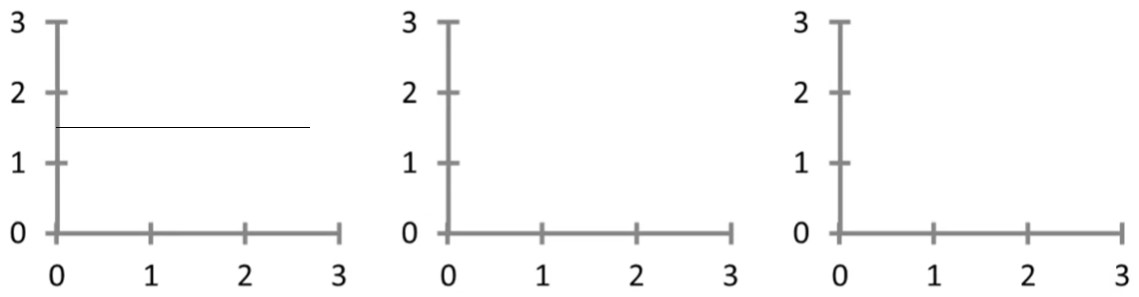
Price

**w**

**0**

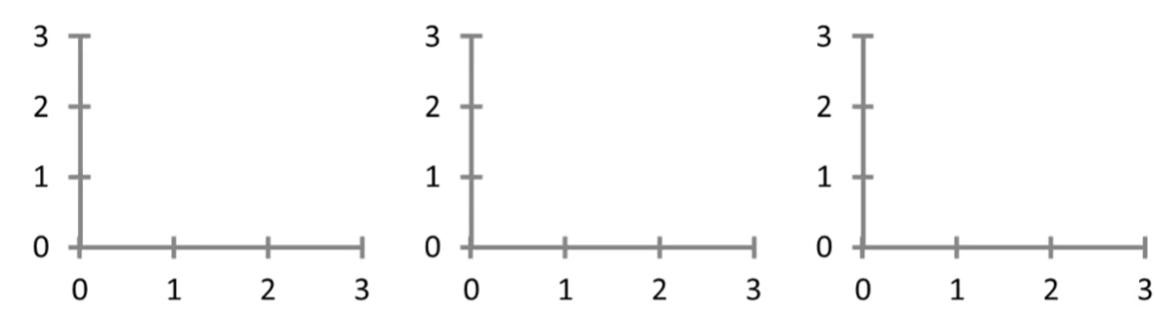
**1**

|  |  |  |
| --- | --- | --- |
| Size | Size | Size |
| w0 = 1.5 w1 = 0 | w0 = 0 w1 = 0.5 | w0 = 1 w1 = 0.5 |



h(x) = 1.5 + 0\*x

Price

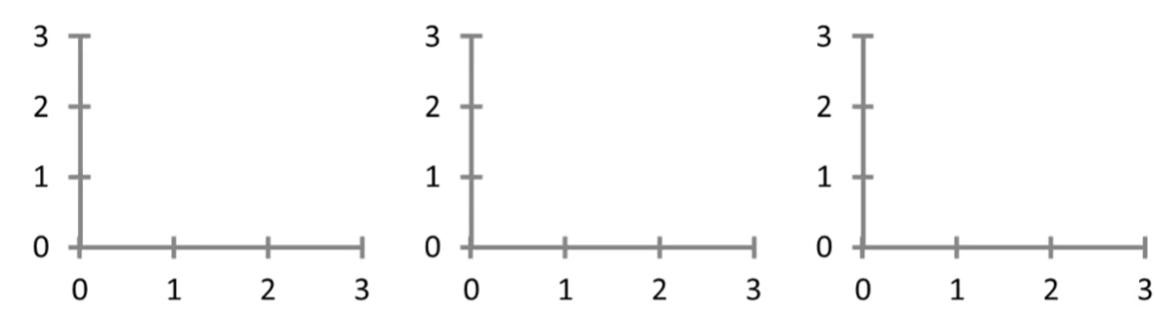


Price

Price

|  |  |  |
| --- | --- | --- |
| Size | Size | Size |
| w0 = 1.5 w1 = 0 | w0 = 0 w1 = 0.5 | w0 = 1 w1 = 0.5 |

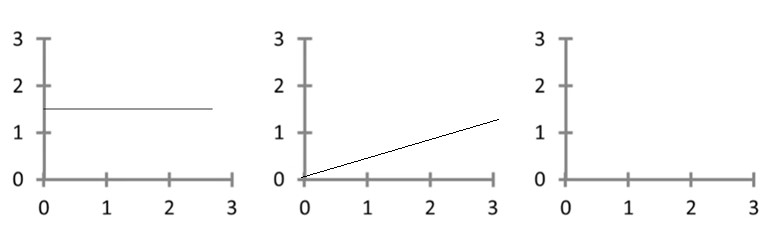
Price



Price

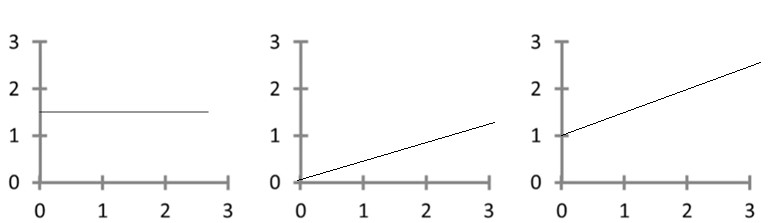
Price

Size Size Size w0 = 1.5 w0 = 0 w0 = 1 w1 = 0 w1 = 0.5 w1 = 0.5



h(x) = 1.5 + 0\*x

h(x) = 0+ 0.5\*x

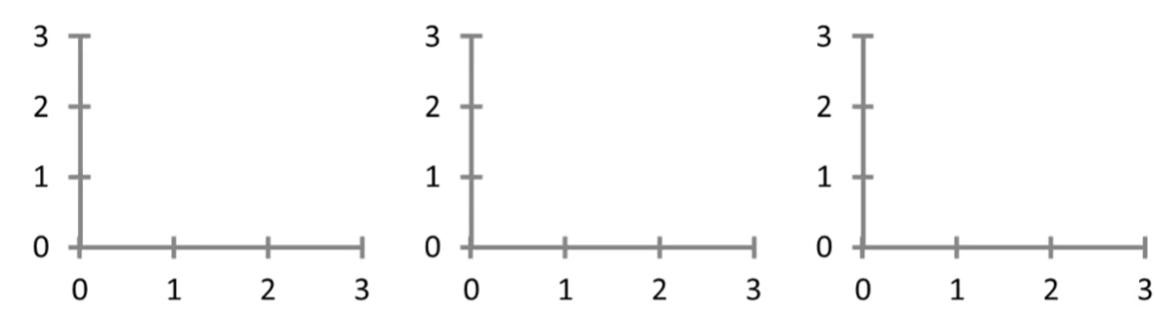


h(x) = 0+ 0.5\*x

h(x) = 1.5 + 0\*x

h(x) = 1 + 0.5\*x

Price

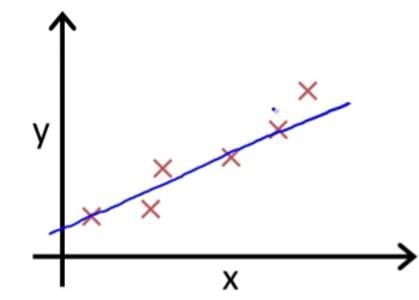
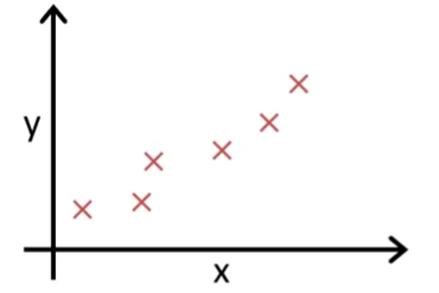


Price

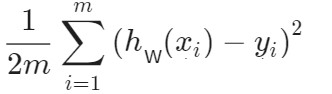
Price

|  |  |  |
| --- | --- | --- |
| Size | Size | Size |
| w0 = 1.5 w1 = 0 | w0 = 0 w1 = 0.5 | w0 = 1 w1 = 0.5 |

**How to choose parameters?**



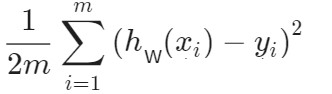
**Idea is** to choose w0 , w1 so that hw(x) is close to y for training examples (x, y)

minimize

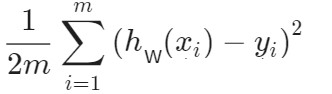
w0, w1 where hw (x) = w0 + w1\*x

# Cost Function

**Cost Function: J(w0 , w1 ):** This takes an average difference of all the results of the hypothesis with inputs from x's and the actual output y's.



**J(w0 , w1 )=**

**Minimize the cost function i.e.,** minimize

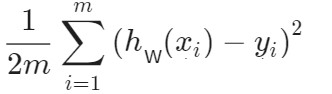
w0, w1

**Hypothesis:**

**hw (x) = w0 + w1\*x**

**Parameters: w0 + w1**

**Cost Function:**

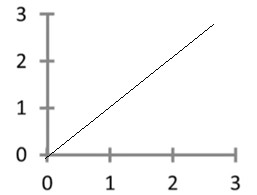
**J(w0 , w1 )=**

**Goal: minimize J(w0, w1)**

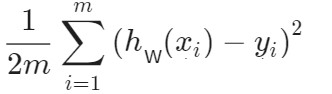
**w0, w1**

Simplified Hypothesis

**hw (x) = w1\* x**

 **w1**

**x**

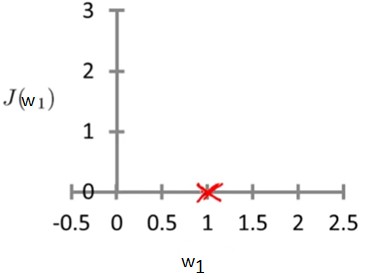
**J(w1 ) =** 

**minimize**

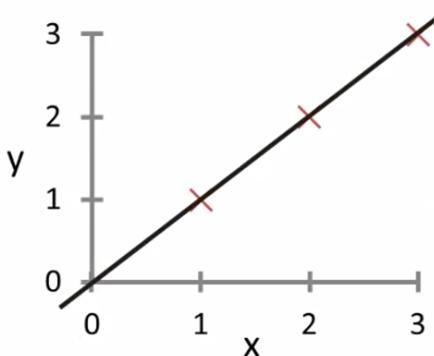
**w0, w1**

## Hypothesis , hw(x) Cost function: J(w1)

(For fixed w1, this is function of x ) (function of parameter w1)

h (x)

w =1



w

1

## w1 =1 w1 =1

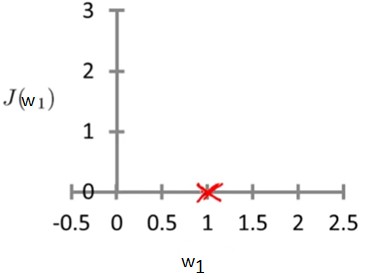
J(1) =(1/2\*3) [((1-1)^2 )+((2-2)^2) + ((3-3)^2)]

J(1)= (1/6) [0^2 +0^2 + 0^2]

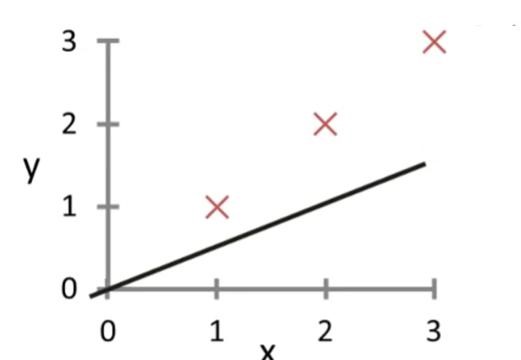
J(1) =(1/6)[0] = 0

## Hypothesis , hw(x) Cost function: J(w1)

(For fixed w1, this is function of x ) (function of parameter w1)

w =1h (x) = 0.5x 

1

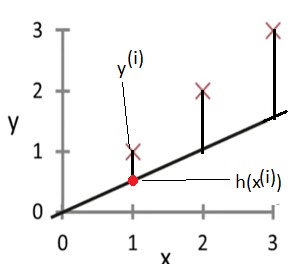
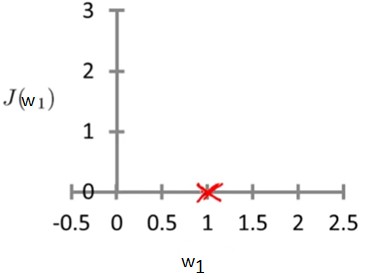


w

**w1 =0.5**

## Hypothesis , hw(x) Cost function: J(w1)

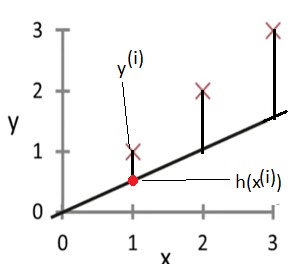
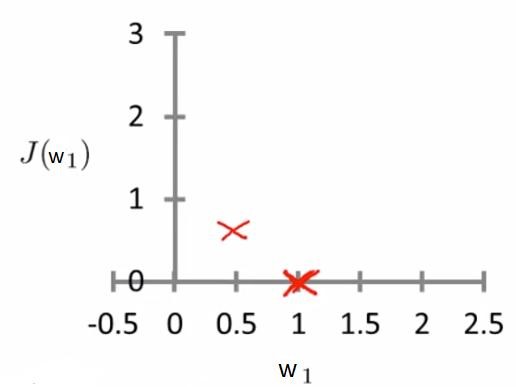
(For fixed w1, this is function of x ) (function of parameter w1)

hw(x) 

**w1 =0.5**

## Hypothesis , hw(x) Cost function: J(w1)

(For fixed w1, this is function of x ) (function of parameter w1)

hw(x) 

## w1 =0.5 w1 =0.5

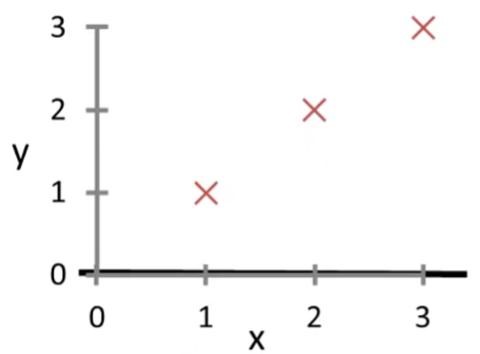
### J(0.5) =(1/2\*3) [((0.5-1)^2 )+((1-2)^2) + ((1.5-3)^2)]

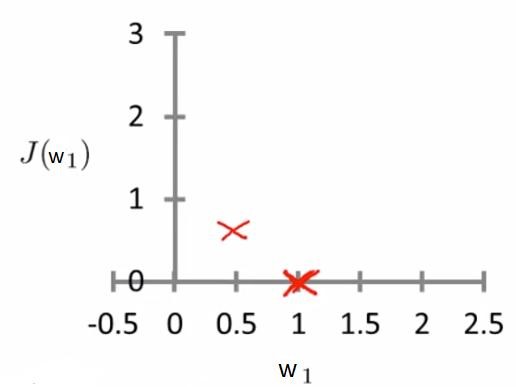
J(0.5)= (1/6) [(-0.5)^2 +(-1)^2 +(-1.5)^2]

J(0.5) =(1/6)[3.5] = 0.58

## Hypothesis , hw(x) Cost function: J(w1)

(For fixed w1, this is function of x ) (function of parameter w1)

 hw(x)=w x

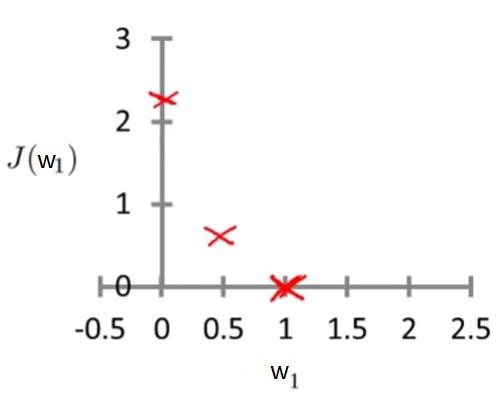
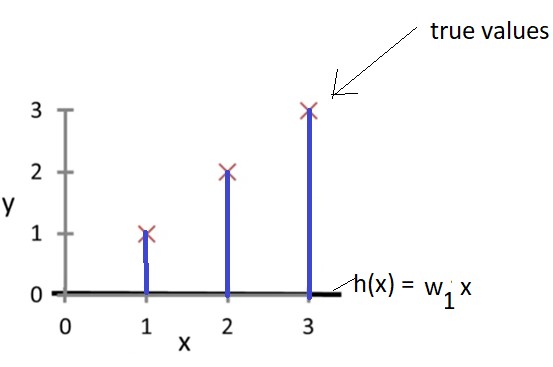


1

**w1 =0**

## Hypothesis , hw(x) Cost function: J(w1)

(For fixed w1, this is function of x ) (function of parameter w1)



## w1 =0 w1 =0

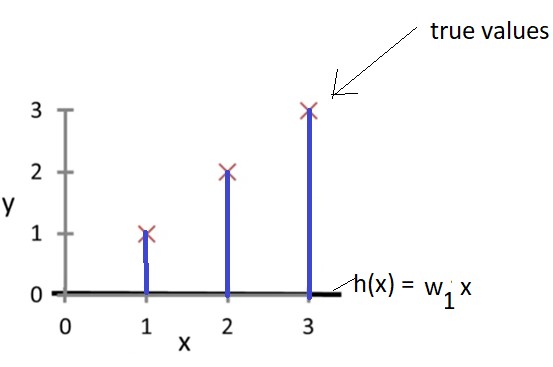
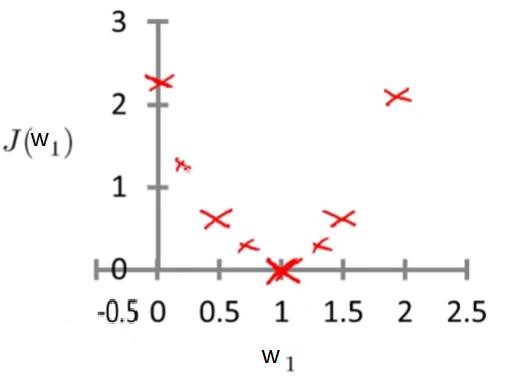
J(0) =(1/2\*3) [((-1)^2 )+((0-2)^2) + ((0-3)^2)]

J(0)= (1/6) [ 1 + 4 + 9 ]

J(0) =(1/6) [14] = 14/6=2.33

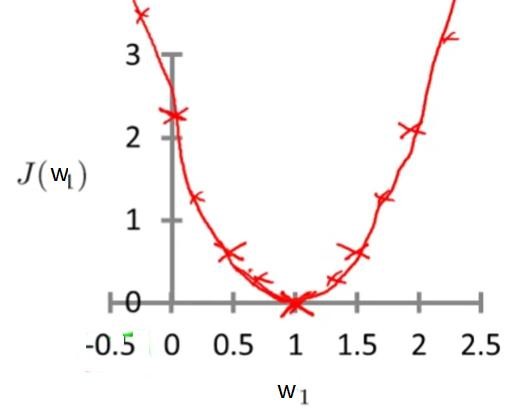
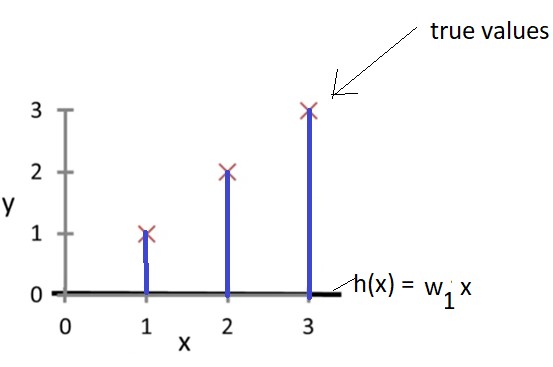
## Hypothesis , hw(x) Cost function: J(w1)

(For fixed w1, this is function of x ) (function of parameter w1)



## w1 =0 w1 =1.5,2,2.5,… Hypothesis , hw(x) Cost function: J(w1)

(For fixed w1, this is function of x ) (function of parameter w1)



**w1 =0 w1 =1.5,2,2.5,…**

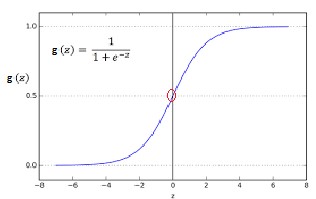
minimize Jw1(W1)

# Logistic Regression

• Logistic Regression is classification problem.

Want : 0<=h(x)<=1

For linear regression: h(x) = WTX

For logistic regression: h(x) = g(WTX) = g(z)= 1/1+e-z – this is logistic function Therefore,

g(z) = 1 . 1+ e-z h(x) = 1 . 1+ e-WTX

# Hypothesis output

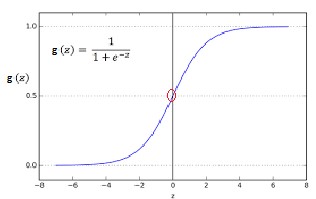
h(x) = estimated probability that y=1 on input x h(x) = 1 . == p(y=1/x,w)

1+ e-WTX

Predict “y=1” if h(x) >=0.5

Predict “y=0” if h(x) < 0.5

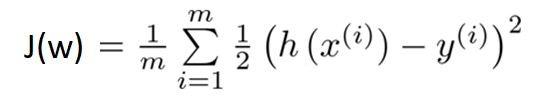
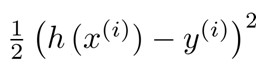
When z>=0, g(z) >=0.5



i.e., h(x)=g(wTx) >=0.5 as z=wTx

# Loss Function

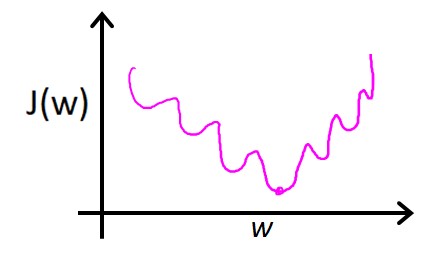
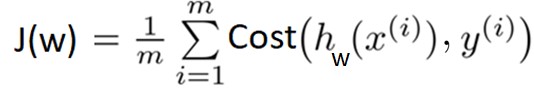
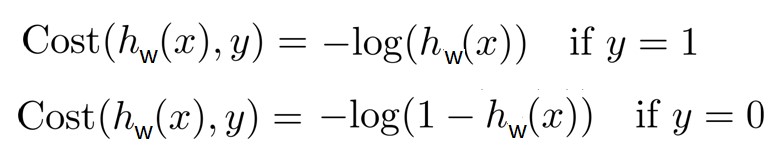
Let, cost(h(x),y)= where, h(x) = 1 .



1+ e-WTX

# Loss/Objective/Error Function

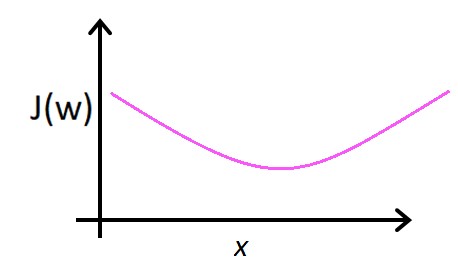
.



Non

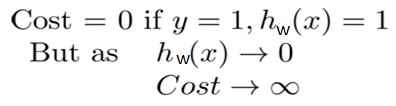
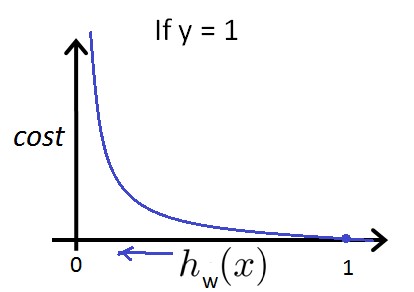
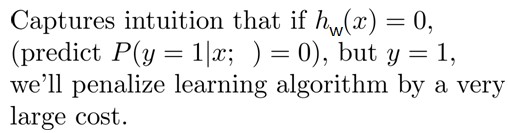
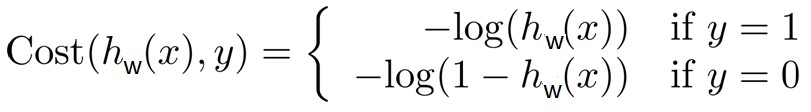
-

convex

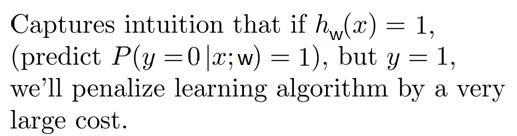
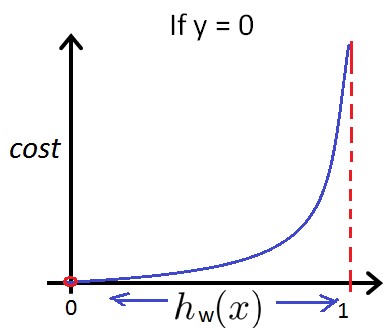
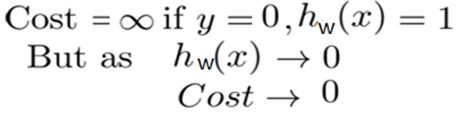
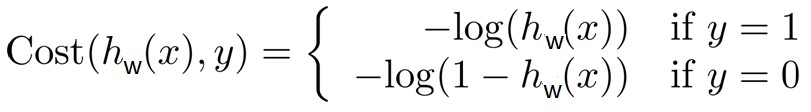


convex

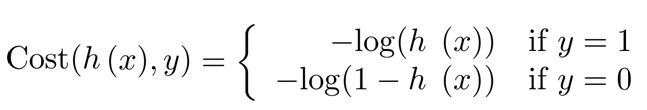
**Logistic regression cost function**

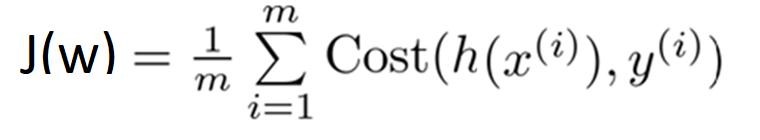


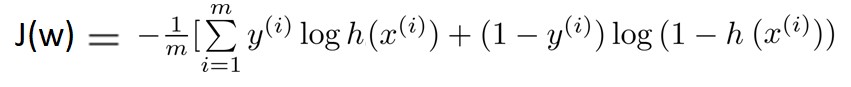
# Logistic regression cost function



## Simplified Logistic regression cost function





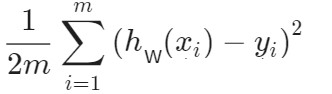


# Problem on Linear Regression

• Given the following data set. Using linear regression, estimate the target variable y as a function of the input feature x. The hypothesis is hw(x) = w0 + w1(x)

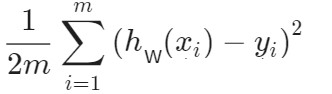
|  |  |
| --- | --- |
| **X** | **Y** |
| 2 | 3 |
| 3 | 4 |
| 4 | 5 |
| 5 | 6 |
| 6 | 7 |

1. Given w parameter , find the ones which will best fit the data: i) w0 = 1, w1 = 0.5 ii) w0 = 1, w1 = 1.5 iii) w0 = 1.5, w1 = 1.
2. Plot the hypothesis for best w parameters and give the value of the cost function for the same, which is mean square error function?
3. Use answer of (1) to evaluate hw(x = 8)

Cost Function: J(w0,w1) =

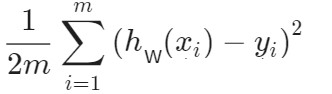
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **w0 = 1 and w1 =0.5** | | | | |
| **X** | **Y** | **h(X) = w0+w1\*X** | **h(X)-Y** | **(h(X)-Y)^2** |
| 2 | 3 | 1 + 0.5 \* 2 = 2 | 2 -3 = -1 | (-1)^2 = 1 |
| 3 | 4 | 1 + 0.5 \* 3 = 2.5 | 2.5 – 4 = -1.5 | (-1.5)^2 = 2.25 |
| 4 | 5 | 1 + 0.5 \* 4= 3 | 3 – 5 = -2 | (-2)^2 = 4 |
| 5 | 6 | 1 + 0.5 \* 5 = 3.5 | 3.5 – 6 = -2.5 | (-2.5)^2 = 6.25 |
| 6 | 7 | 1 + 0.5 \* 6 = 4 | 4 – 7 = -3 | (-3)^2 = 9 |
|  |  |  |  | Σ**(h(X)-Y)^2 = 22.5** |

J(1,0.5) = (1/2\*5) \*22.5 = (1/10) \* 22.5 = 2.25

Cost Function: J(w0,w1) =

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **w0 = 1 and w1 =1.5** | | | | |
| **X** | **Y** | **h(X) = w0+w1\*X** | **h(X)-Y** | **(h(X)-Y)^2** |
| 2 | 3 | 1 + 1.5 \* 2 = 4 | 4 -3 = 1 | (1)^2 = 1 |
| 3 | 4 | 1 + 1.5 \* 3 = 5.5 | 5.5 – 4 = 1.5 | (1.5)^2 = 2.25 |
| 4 | 5 | 1 + 1.5 \* 4= 7 | 7 – 5 = 2 | (2)^2 = 4 |
| 5 | 6 | 1 + 1.5 \* 5 = 8.5 | 8.5 – 6 = 2.5 | (2.5)^2 = 6.25 |
| 6 | 7 | 1 + 1.5 \* 6 = 10 | 10 – 7 = 3 | (3)^2 = 9 |
|  |  |  |  | Σ**(h(X)-Y)^2 = 22.5** |

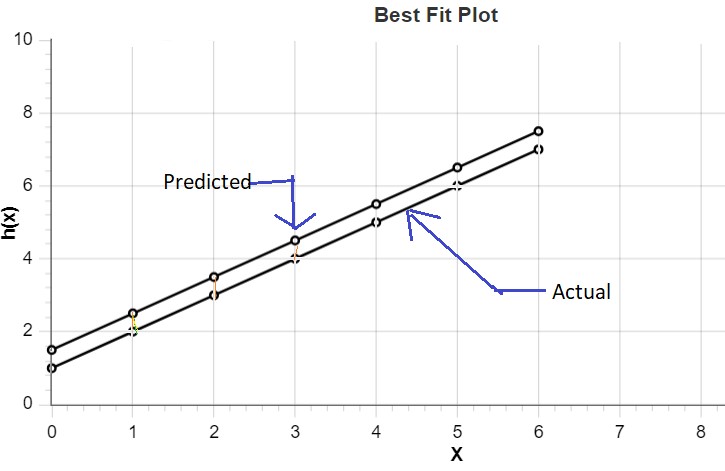
J(1,1.5) = (1/2\*5) \*22.5 = (1/10) \* 22.5 = 2.25

Cost Function: J(w0,w1) =

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **w0 = 1.5 and w1 =1** | | | | |
| **X** | **Y** | **h(X) = w0+w1\*X** | **h(X)-Y** | **(h(X)-Y)^2** |
| 2 | 3 | 1.5 + 1 \* 2 = 3.5 | 3.5 - 3 = 0.5 | (0.5)^2 = 0.25 |
| 3 | 4 | 1.5 + 1 \* 3 = 4.5 | 4.5 – 4 = 0.5 | (0.5)^2 = 0.25 |
| 4 | 5 | 1.5 + 1 \* 4= 5.5 | 5.5 – 5 = 0.5 | (0.5)^2 = 0.25 |
| 5 | 6 | 1.5 + 1 \* 5 = 6.5 | 6.5 – 6 = 0.5 | (0.5)^2 = 0.25 |
| 6 | 7 | 1.5 + 1 \* 6 = 7.5 | 7.5 – 7 = 0.5 | (0.5)^2 = 0.25 |
|  |  |  |  | Σ**(h(X)-Y)^2 = 1.25** |

J(1.5,1) = (1/2\*5) \*1.25 = (1/10) \* 1.25 = 0.125

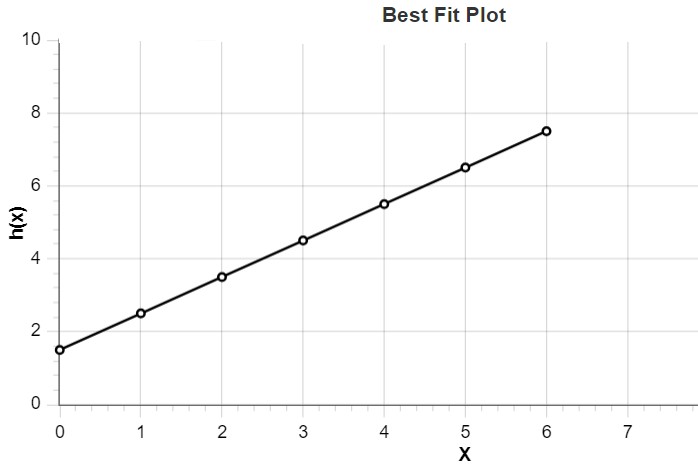
Plot of Difference between true and predicted values for w0=1.5 and w1=1



Solution

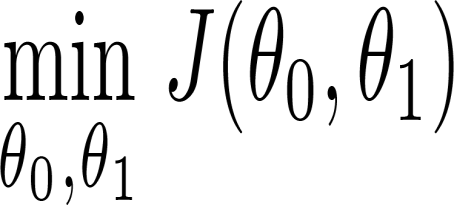
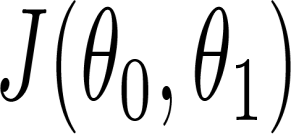
|  |  |  |
| --- | --- | --- |
| **w0** | **w1** | **J(w0 , w1 )** |
| 1 | 0.5 | 2.25 |
| 1 | 1.5 | 2.25 |
| **1.5** | **1** | **0.125** |

Cost is minimum for the weight values w0=1.5 and w1=1. these are the parameters which best fit the data. Plot is:



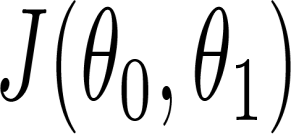
# Gradient Descent

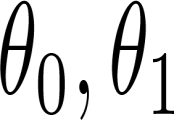
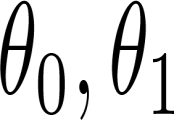
Have some function



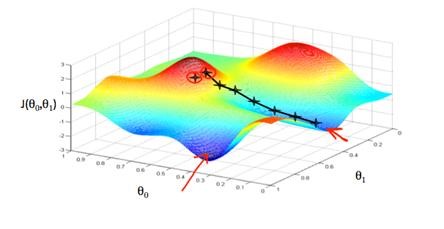
Want

**Outline:**

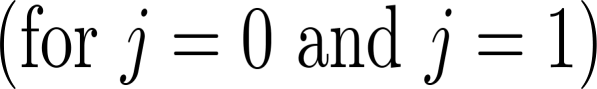
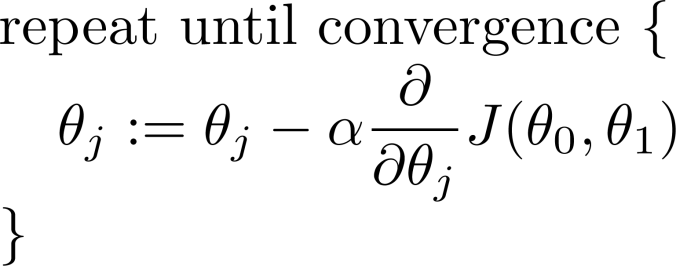
* Start with some
* Keep changing to reduce until we hopefully end up at a minimum



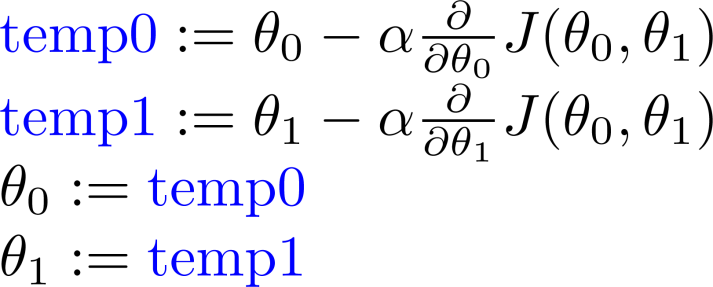
# Gradient Descent



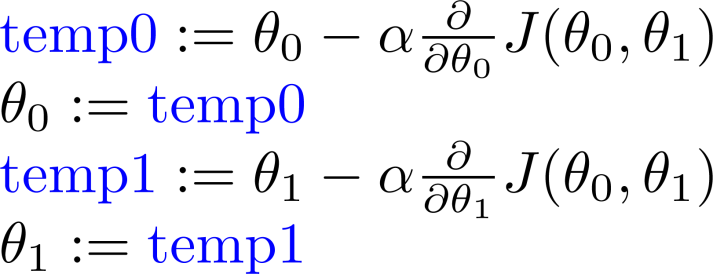
## Gradient descent algorithm



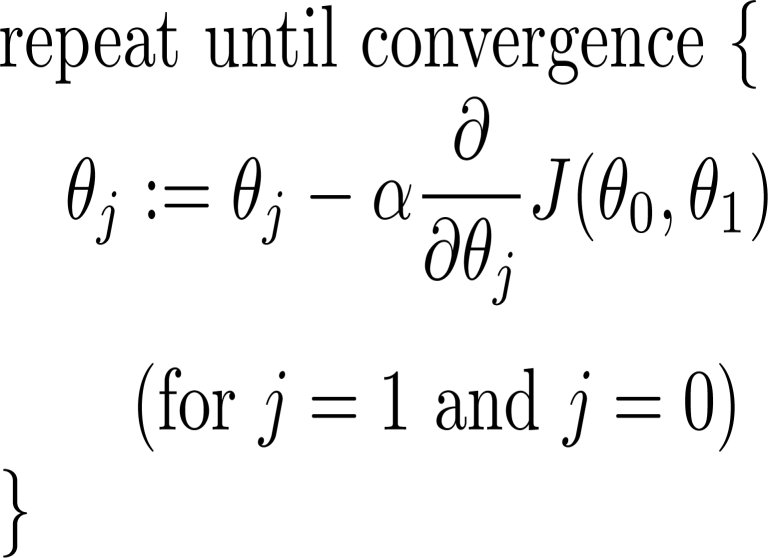
Correct: Simultaneous update

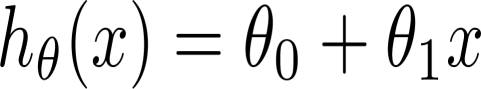


Incorrect:

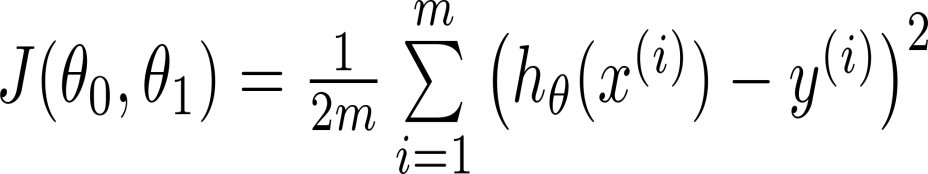


## Linear regression with one variable Gradient Descent

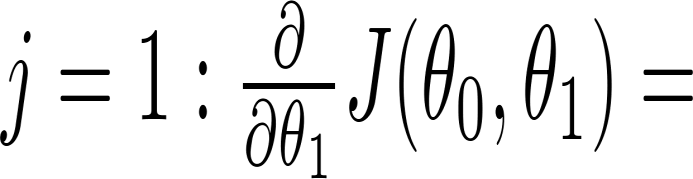
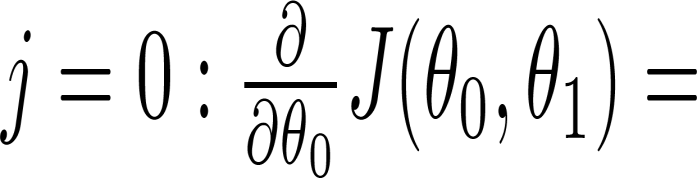
**Gradient descent algorithm**



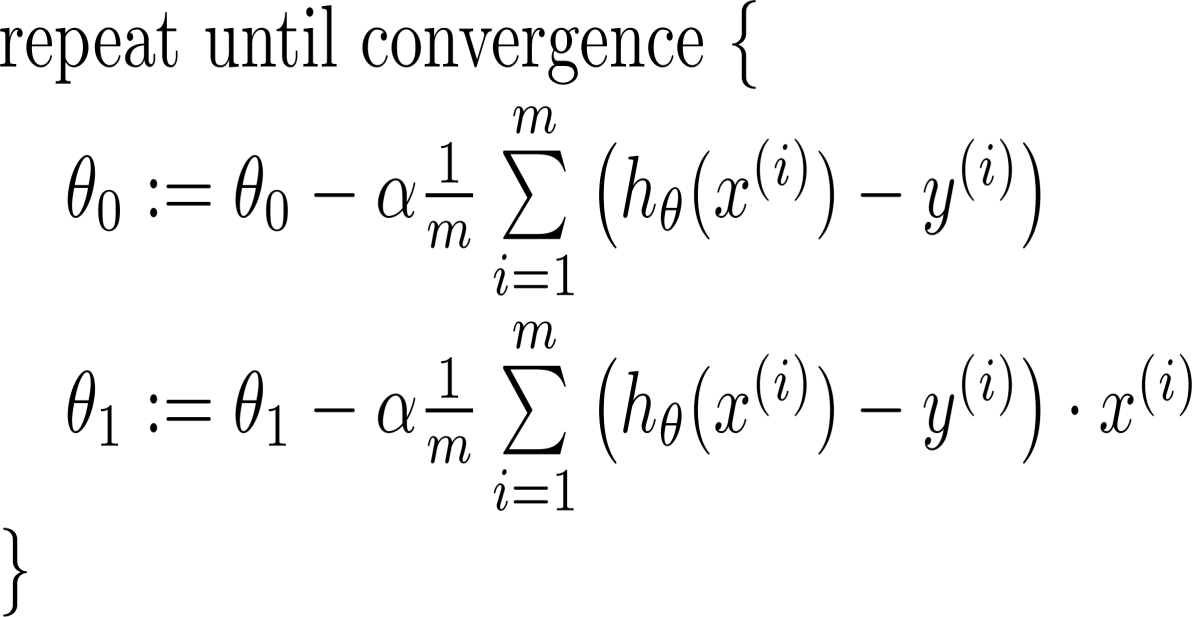
**Linear Regression Model**







### **Gradient descent algorithm**



update

and

simultaneously