MACHINE LEARNING: LINEAR REGRESSION

UNIVERSIDAD DE ANTIOQUIA

1803

UNIVERSIDAD DE ANTIOQUIA 1803

Dataset split

Size of the house (m^2)	Price (millions)
20	60
30	100
40	120
50	180
60	200
70	250
80	300
90	350
100	450
110	480

Training set
70%

Test set
30%

Training/cross validation/test set

UNIVERSIDAD DE ANTIOQUIA

1803

Dataset split

Size of the house (m^2)	Price (millions)
20	60
30	100
40	120
50	180
60	200
70	250
80	300
90	350
100	450
110	480

Training set
60%

Cross validation set (dev set)
20%

Test set
20%

Cross-validation

https://www.youtube.com/watch?v=TIgfjmp-4BA

Background-What is Machine Learning?

UNIVERSIDAD DE ANTIOQUIA

1803

Applications

- Website search (group website pages)
- Streaming movies (similar movies recommendation)
- OK Google, Siri, Alexa
- Email classification (spam, no spam)
- Healthcare
- Agriculture
- ……..

MACHINE LEARNING

"Field of study that gives computers the ability to learn without being explicitly programmed" Arthur Samuel (1959).

Machine Learning Algorithms

UNIVERSIDAD DE ANTIOQUIA
1803

- Supervised Learning
- Unsupervised Learning

Supervised Learning

- The algorithm learns from x to y mapping.

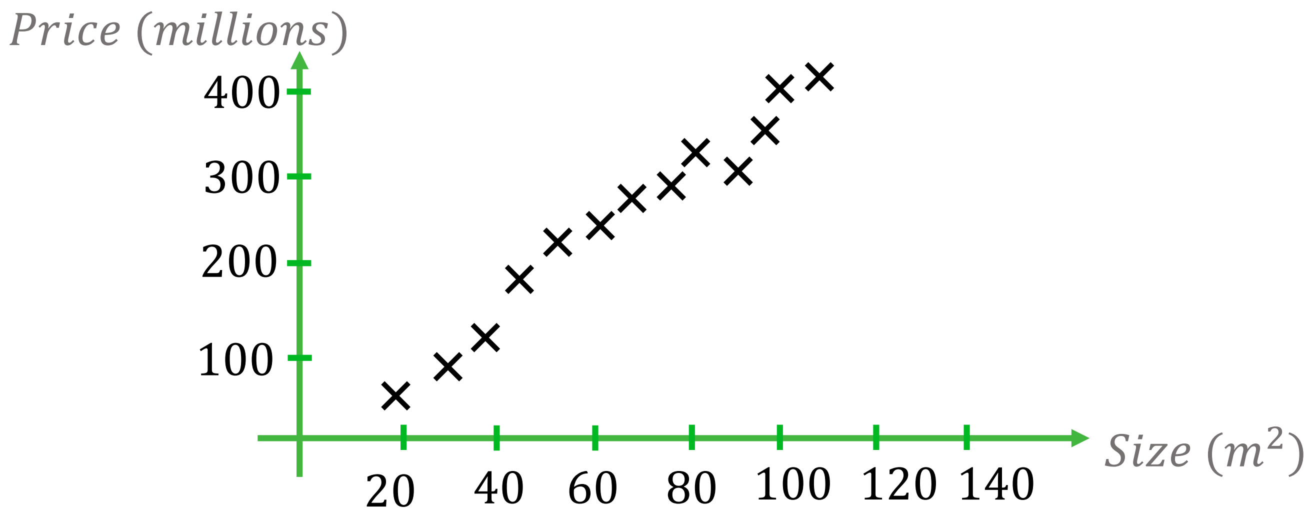
$$\begin{array}{ccc} x & \rightarrow & y \\ \text{Input} & & \text{Output} \end{array}$$

- The algorithm takes examples to learn from. It has to know the “correct labels”.

Supervised Learning

UNIVERSIDAD DE ANTIOQUIA

- Regression: house price prediction



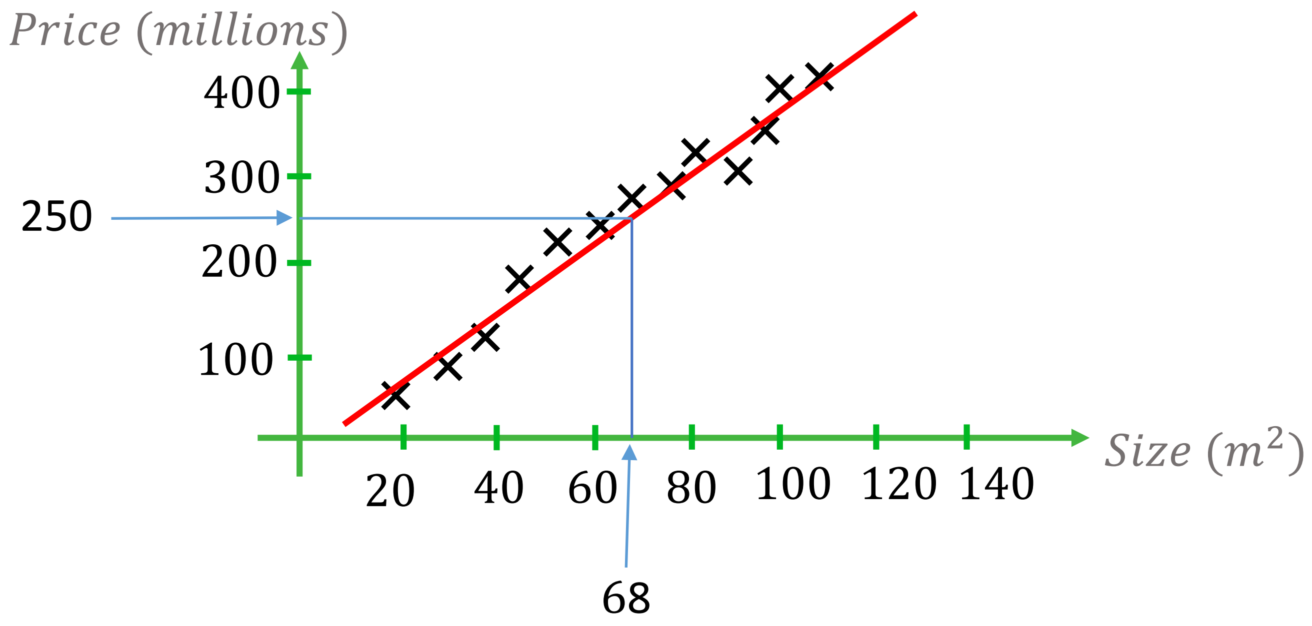
Supervised Learning

UNIVERSIDAD DE ANTIOQUIA

1803

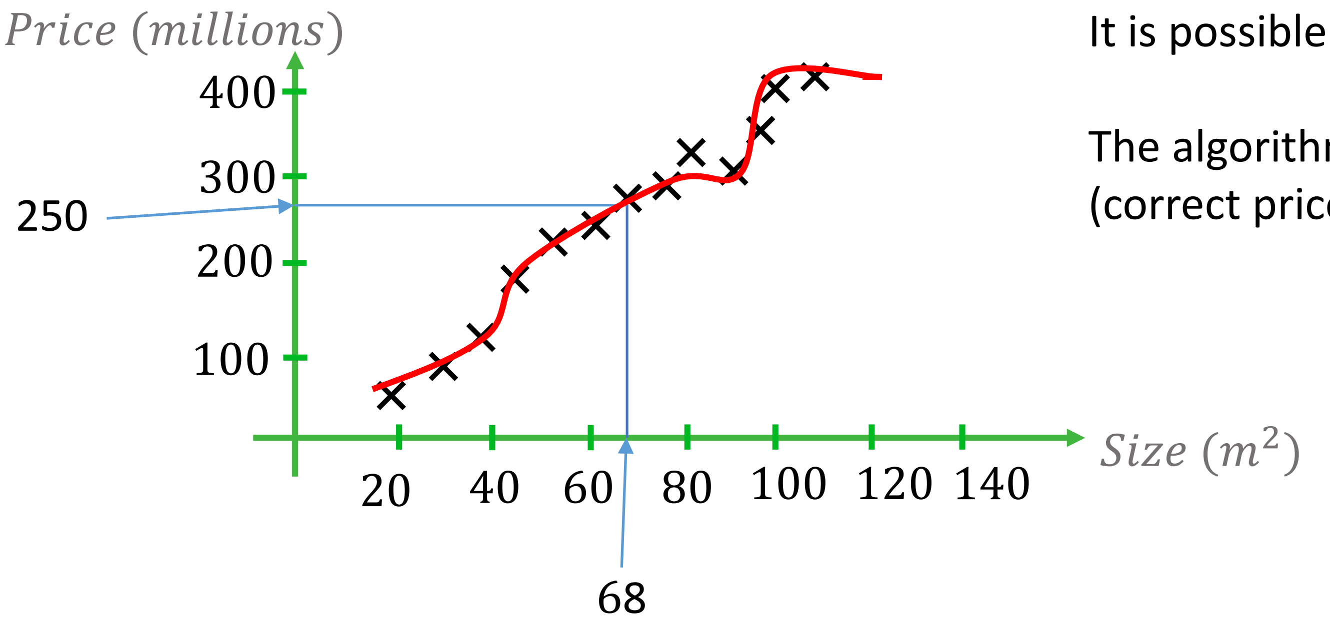
- Regression

It is possible to fit a straight line or a curve



Supervised Learning

- Regression



It is possible to fit a straight line or a curve

The algorithm has to know the “right answer” (correct price)

- **Regression**

It predicts a number

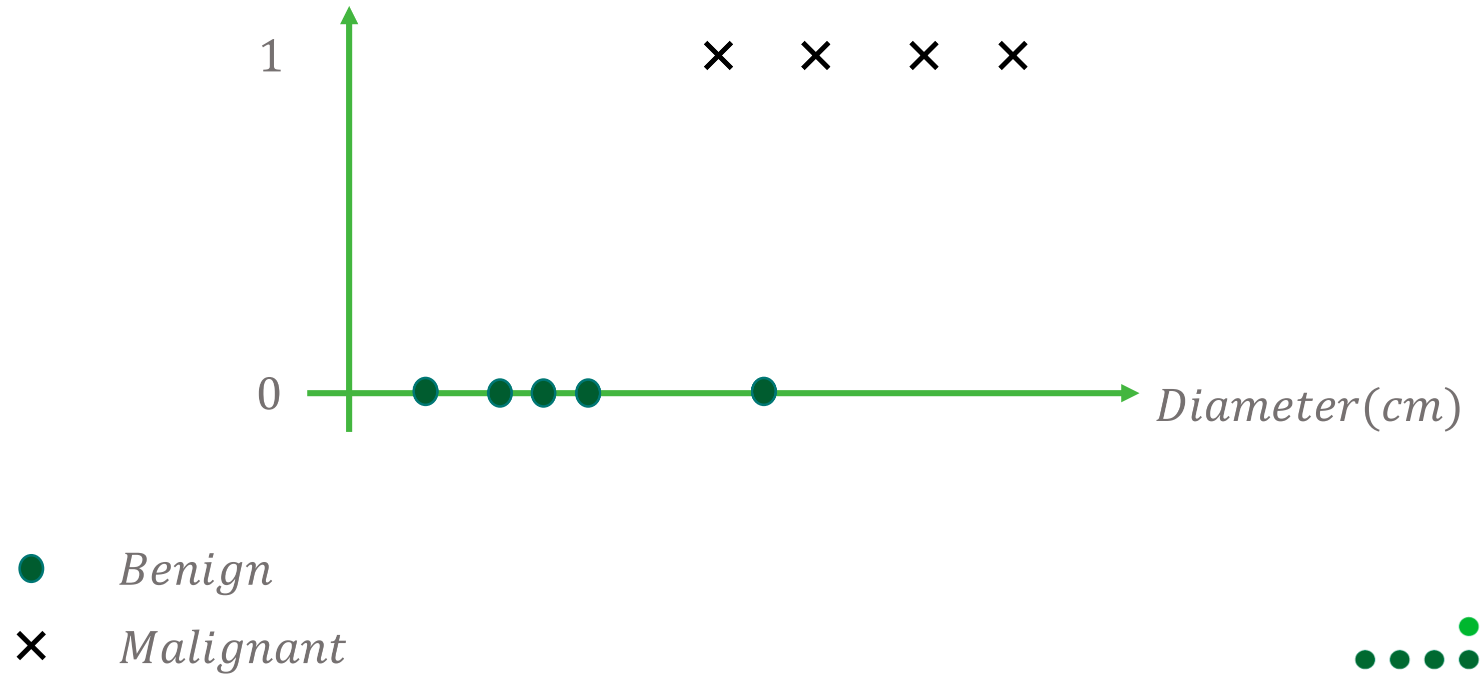
There are many possible outputs

- **Classification**

It predicts a small number of possible outputs or categories (classes)

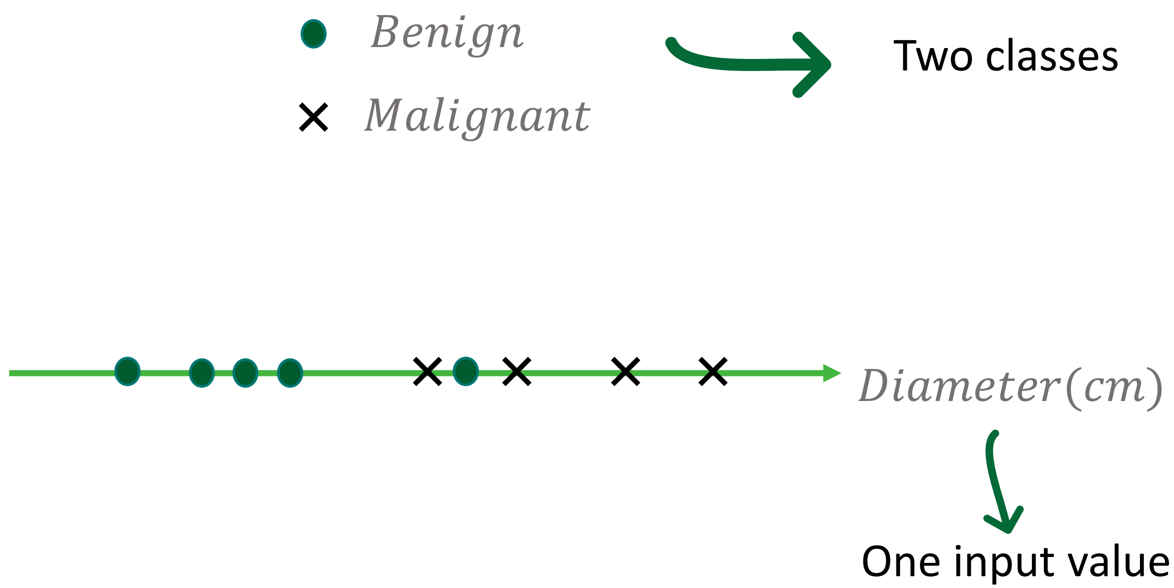
Categories or classes do not have to be numbers. For example, classify between cats and dogs, classify between malignat or benign for a tumor

- Classification: melanoma detection



Supervised Learning

- Classification: melanoma detection



UNIVERSIDAD DE ANTIOQUIA

- **Classification: melanoma detection**

$Benign$

$\times$ $Malignant: type\ I$

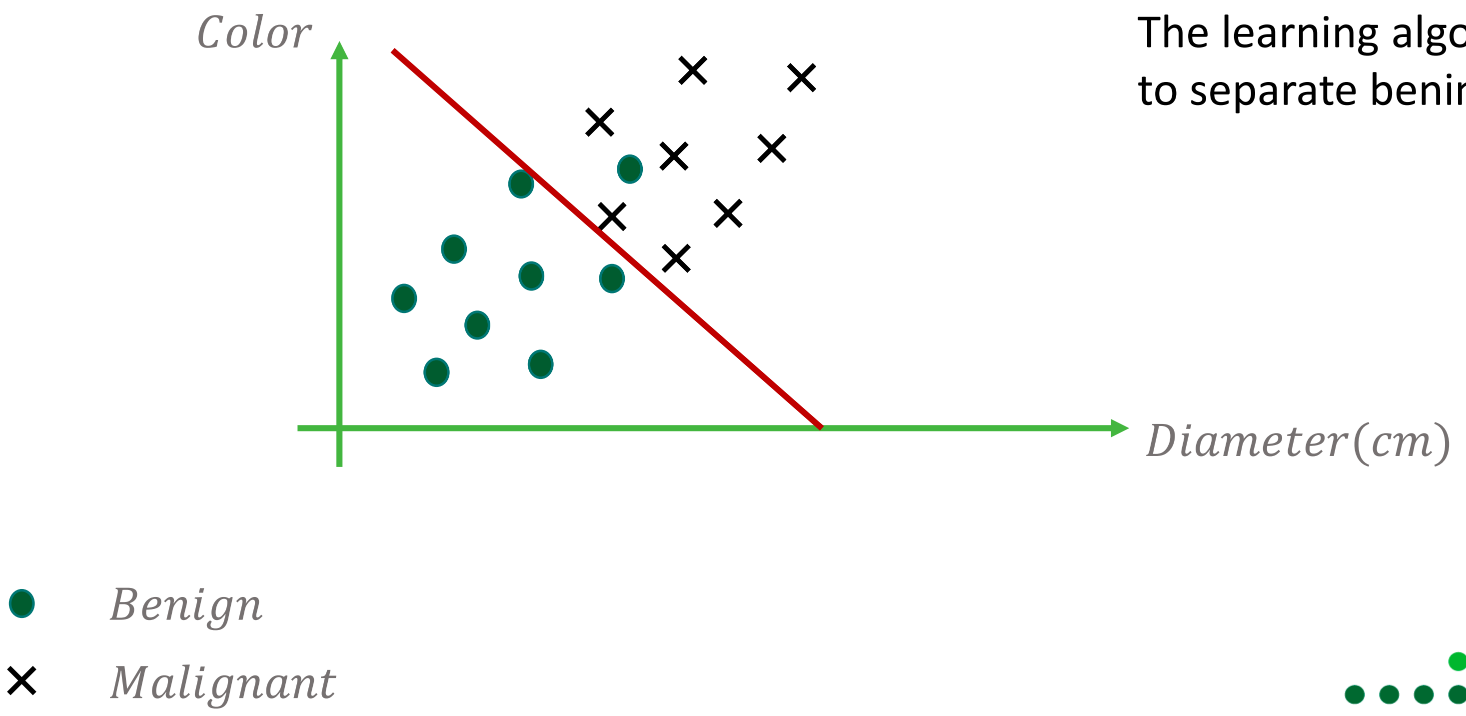
$Malignant: type\ II$

Three classes

$Diameter(cm)$

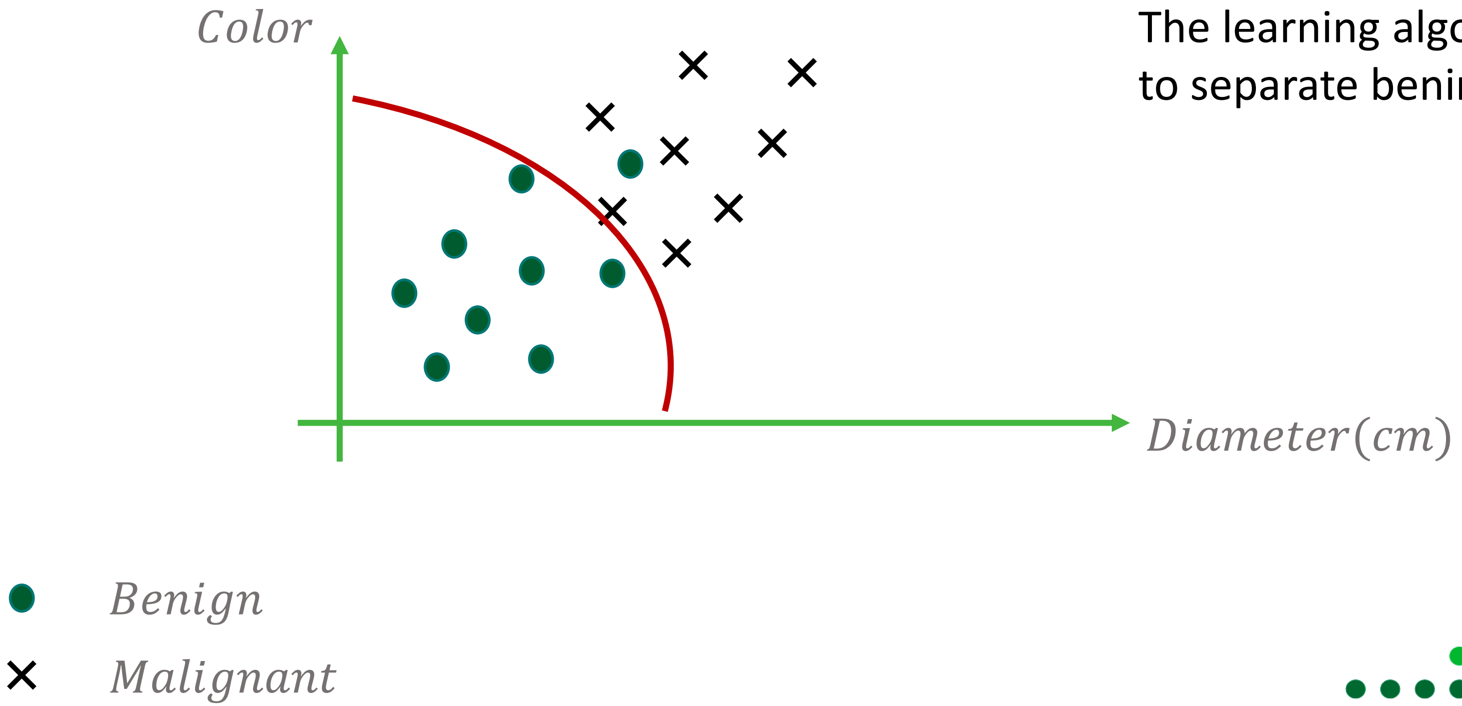
One input value

▪Classification: melanoma detection



The learning algorithm finds a **boundary** to separate bening and malignant tumors

▪Classification: melanoma detection



The learning algorithm finds a **boundary** to separate bening and malignant tumors

Supervised Learning

Regression

Classification

Unsupervised Learning

- Data is not associated with any labels or classes or categories.

- The objective is to find some structure in the data, something interesting.

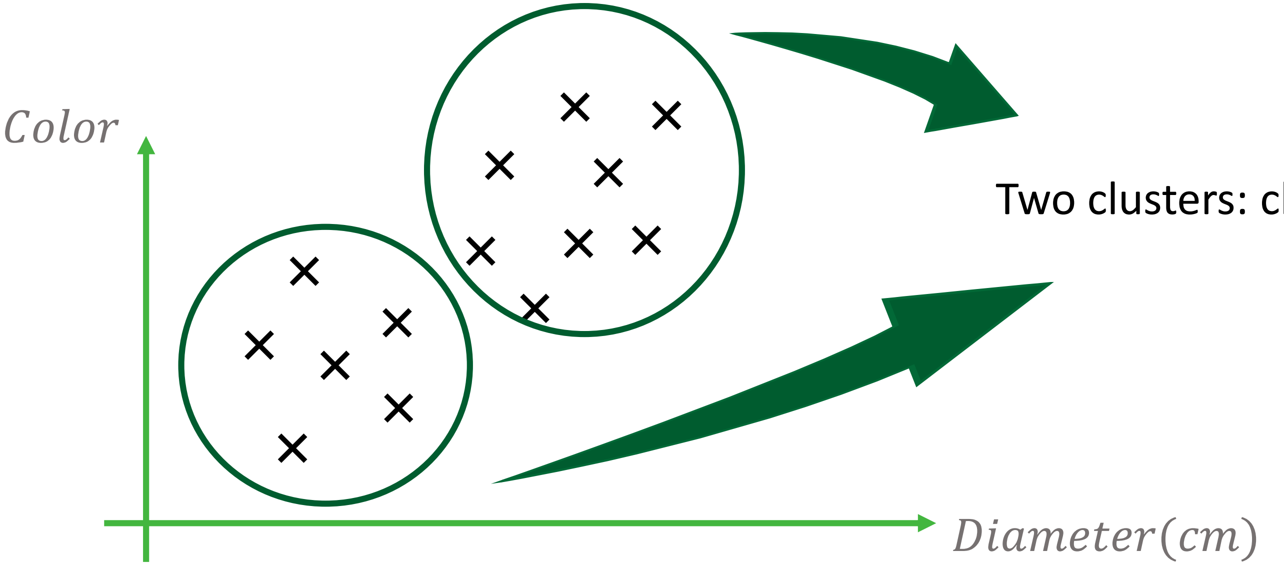
Unsupervised Learning

Find some structure un unlabeled data

Unsupervised Learning

UNIVERSIDAD DE ANTIOQUIA
1803

Find some structure un unlabeled data



Two clusters: clustering (group similar data)

Example: Google search about a topic

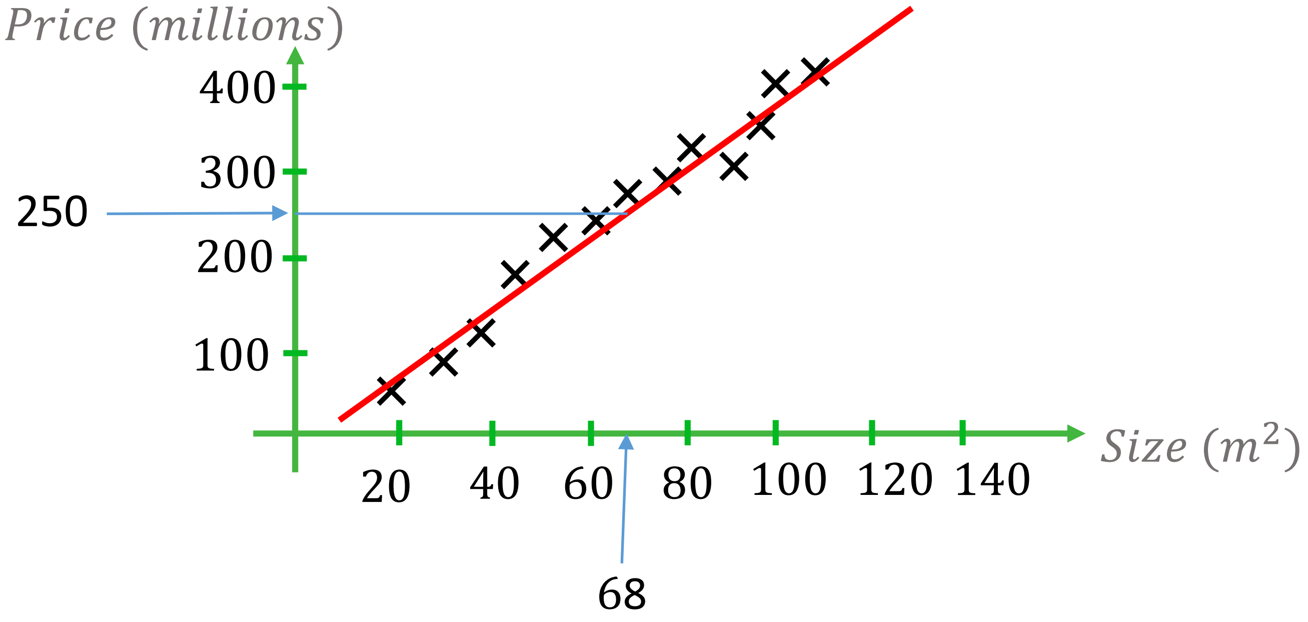
Unsupervised Learning

- Clustering
- Anomaly detection: it finds unusual data
- Dimensionality reduction: PCA

Linear Regression

- It fits a straight line to the data

- Linear Regression: house price prediction



Size (m^2)	Price (millions)
20	60
30	100
40	120
50	180
…	…

Supervised Algorithm: Linear Regression

UNIVERSIDAD DE ANTIOQUIA

1803

Terminology

- **Training set:** data that is used to train the model
  - $x \longrightarrow$ input variable (feature)
  - $y \longrightarrow$ output variable
  - $m \longrightarrow$ number of training examples
  - $(x, y) \longrightarrow$ a single training example

Size (m^2)	Price (millions)
x	y
20	60
30	100
40	120
50	180
…	…

Supervised Algorithm: Linear Regression

UNIVERSIDAD DE ANTIOQUIA

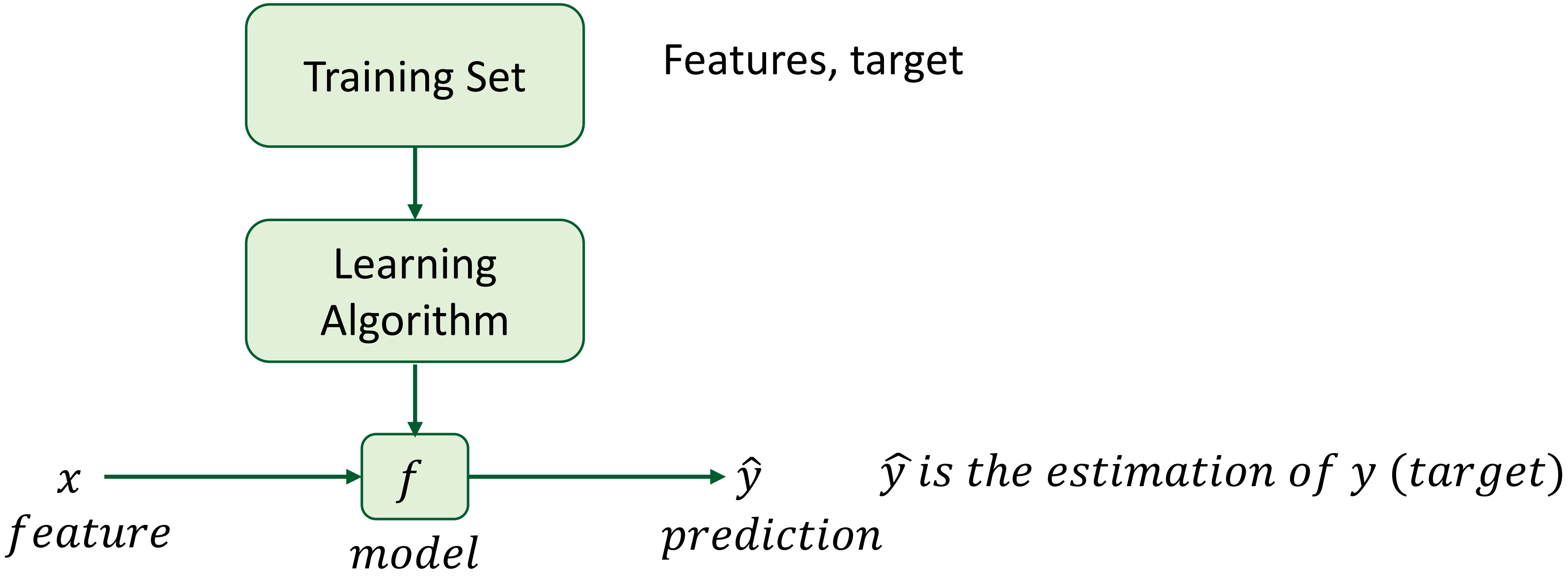
1803

Terminology

- **Training set:** data that is used to train the model
  - $(x^{(i)}, y^{(i)}) \longrightarrow i^{th}$ training example
  - i refers to the rows in the table
  - $(x^{(3)}, y^{(3)}) = (40, 120)$

Size (m^2)	Price (millions)
x	y
20	60
30	100
40	120
50	180
…	…

Supervised Algorithm: Linear Regression



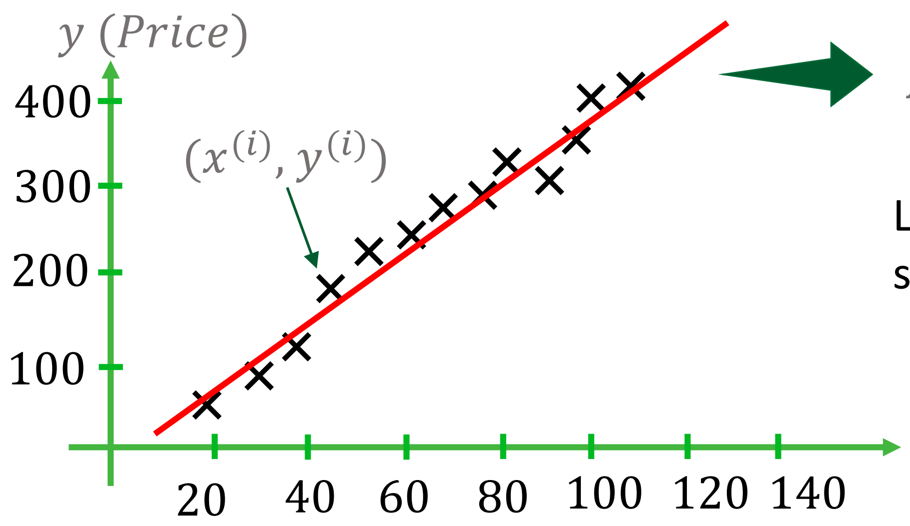
House Price prediction

f model representation:

$$f_{w,b}(x) = wx + b \longrightarrow f(x) = wx + b$$

The values of w and b are going to determine the prediction $\hat{y}$

- Linear Regression: house price prediction



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

Linear regression with one variable
single feature (x)

Cost Function:

It tells how well the model is working, so the idea is to get better results based on it.

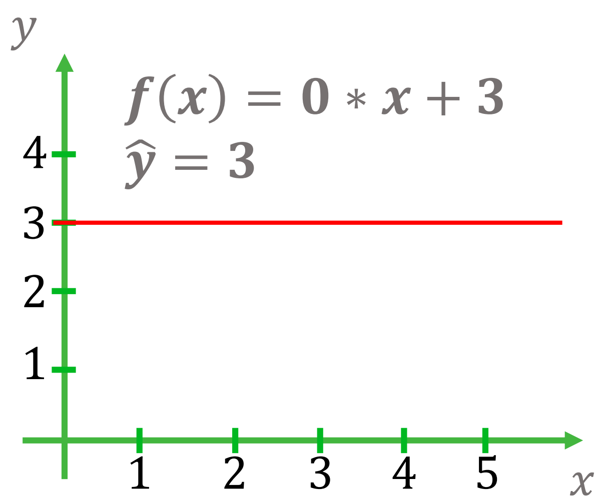
For the model:

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

$\boldsymbol{w}, \boldsymbol{b}$ are called parameters, coefficients or weights

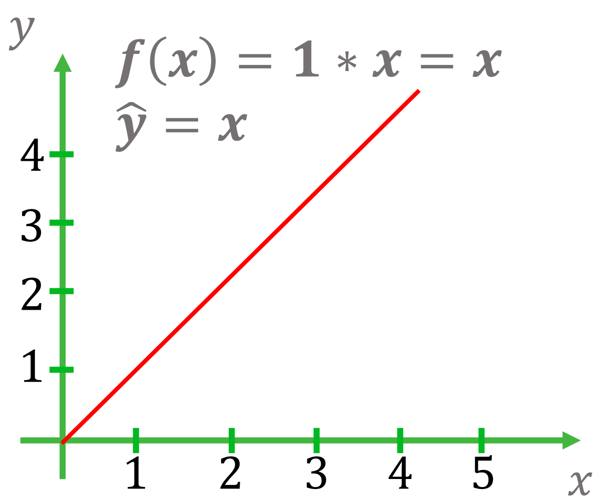
Supervised Algorithm: Linear Regression with one variable

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



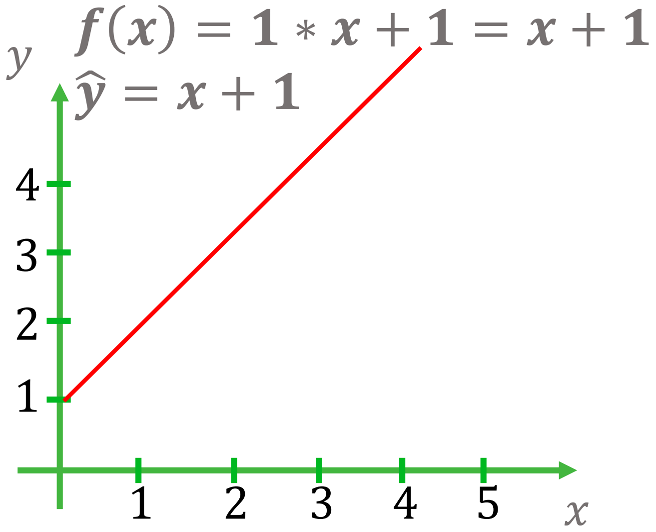
$w = 0, slope$

$b = 3, intercept$



$w = 1, slope$

$b = 0, intercept$



$w = 1, slope$

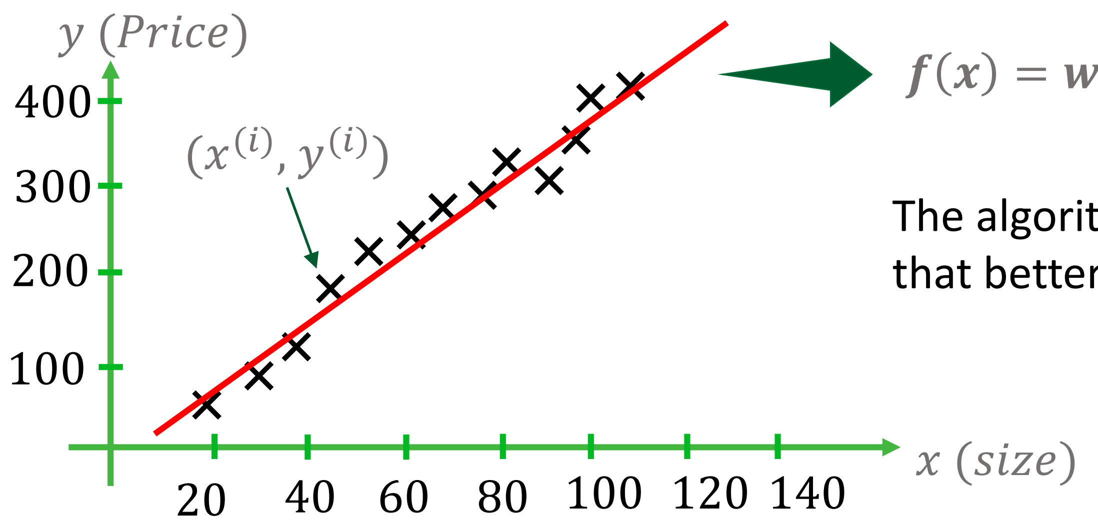
$b = 1, intercept$

Supervised Algorithm: Linear Regression

UNIVERSIDAD DE ANTIOQUIA

1803

- Linear Regression: house price prediction



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

The algorithm finds the $\boldsymbol{w}$ and $\boldsymbol{b}$ (parameters) that better fits the data

$\hat{y}^{(i)}=f_{w,b}\left(x^{(i)}\right)$

$f_{w,b}\left(x^{(i)}\right) = w\, x^{(i)} + \mathrm{b}$

UNIVERSIDAD DE ANTIOQUIA
1803

How to find w and b values so,

$\hat{y}^{(i)}$ is close to $y^{(i)}$ for all $(x^{(i)}, y^{(i)})$

?

Cost Function: squared error cost function

$$J(w,b) = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y}^{(i)} - y^{(i)})^2$$

$m: number\ of\ training\ examples$

UNIVERSIDAD DE ANTIOQUIA

1803

Cost Function: squared error cost function

$$J(w,b) = \frac{1}{2m}\sum_{i=1}^{m}\underbrace{\left(\hat{y}^{(i)} - y^{(i)}\right)^2}_{\text{Error}}$$

m: *number of training examples*

$\hat{y}^{(i)}$: prediction

$y^{(i)}$: target

UNIVERSIDAD DE ANTIOQUIA

1803

Cost Function: squared error cost function

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

$m: number\ of\ training\ examples$

$f_{w,b}\left(x^{(i)}\right)$: prediction

$y^{(i)}$: target

Supervised Learning: Linear Regression

Model: $f_{w,b}(x) = wx + b$

Parameters: w, b

Cost function: $J(w, b) = \frac{1}{2m} \sum_{i=1}^{m} \left(f_{w,b}\left(x^{(i)}\right) - y^{(i)}\right)^2$

Goal: $\underset{w,b}{minimize}\, J(w, b)$

Supervised Learning: Linear Regression

UNIVERSIDAD DE ANTIOQUIA

1803

If b=0,

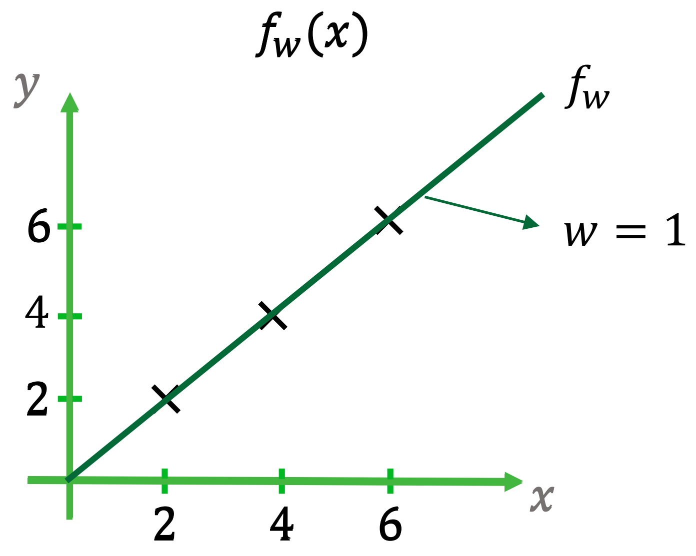
Model: $f_w(x) = wx$

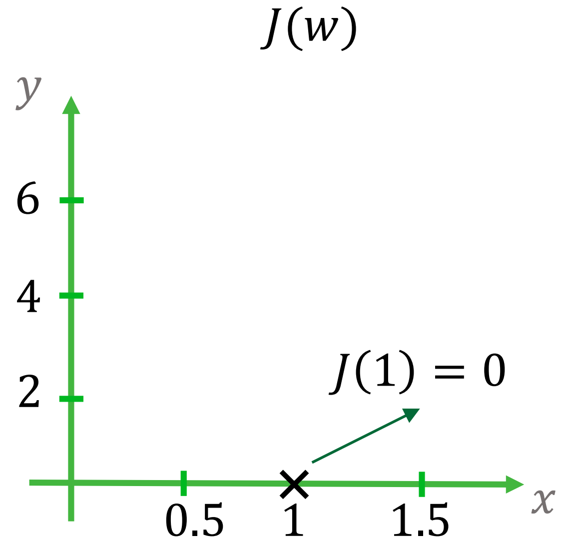
Parameters: w

Cost function: $J(w) = \frac{1}{2m}\sum_{i=1}^{m}(f_w\left(x^{(i)}\right) - y^{(i)})^2$

Goal: $\underset{w}{minimize}\, J(w)$

Supervised Learning: Linear Regression





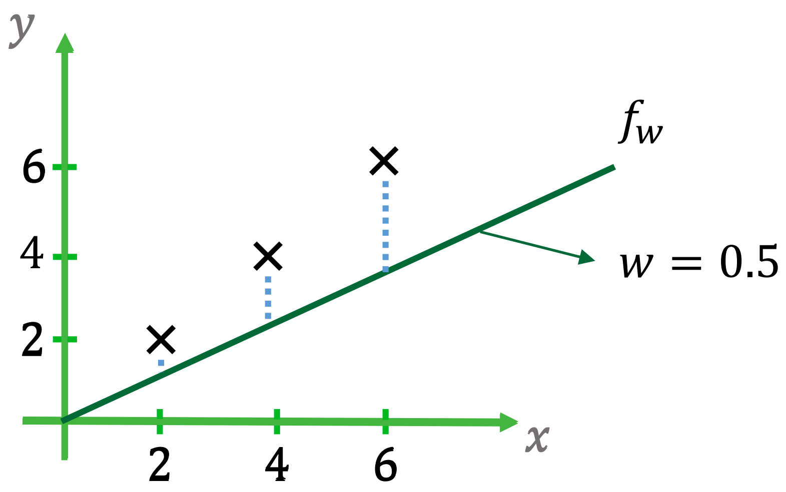
$$w = 1$$

$$J(w) = \frac{1}{2m}(0^2 + 0^2 + 0^2) = 0$$

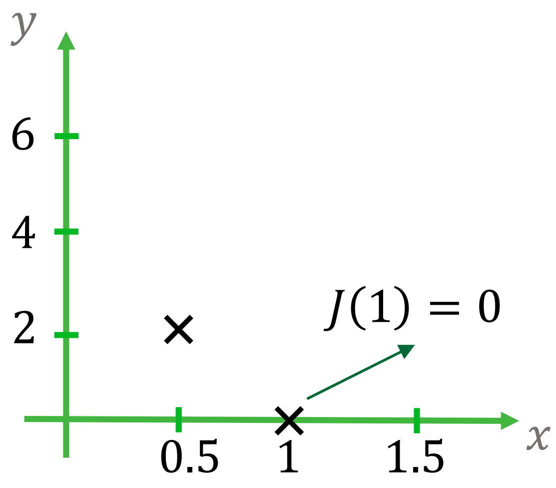
$$J(1) = 0$$

Supervised Learning: Linear Regression

$f_w(x)$



$J(w)$



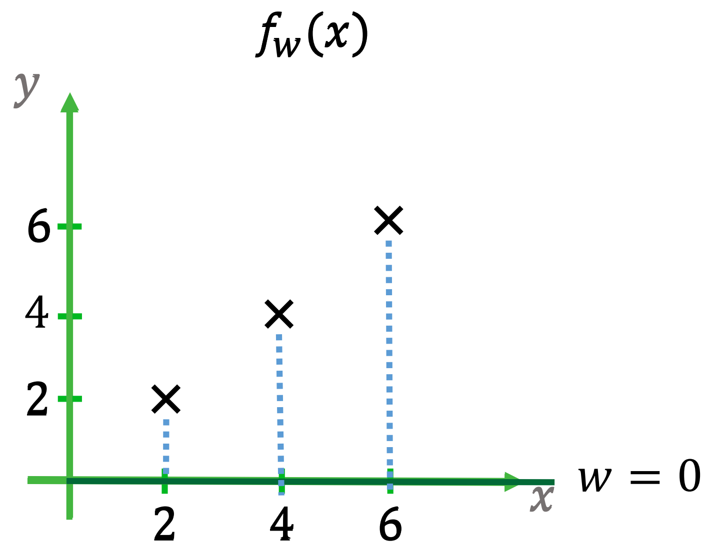
$w = 0.5$

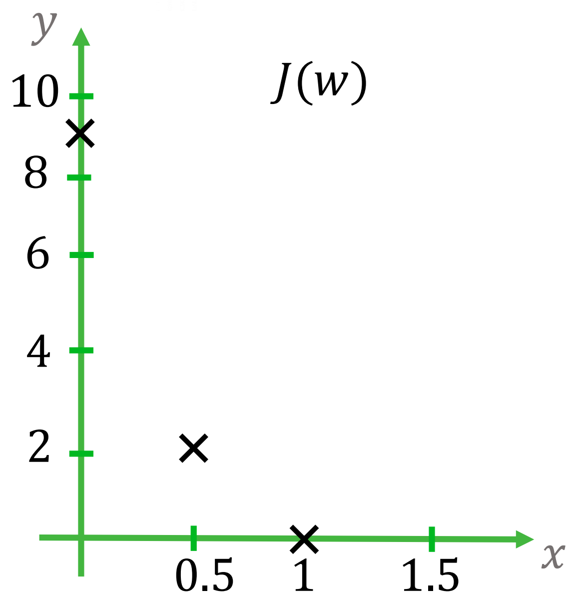
$$J(0.5) = \frac{1}{2 * 3}\left((1-2)^2 + (2-4)^2 + (3-6)^2\right) = 2.3$$

$$J(0.5) = 2.3$$

Supervised Learning: Linear Regression

UNIVERSIDAD DE ANTIOQUIA





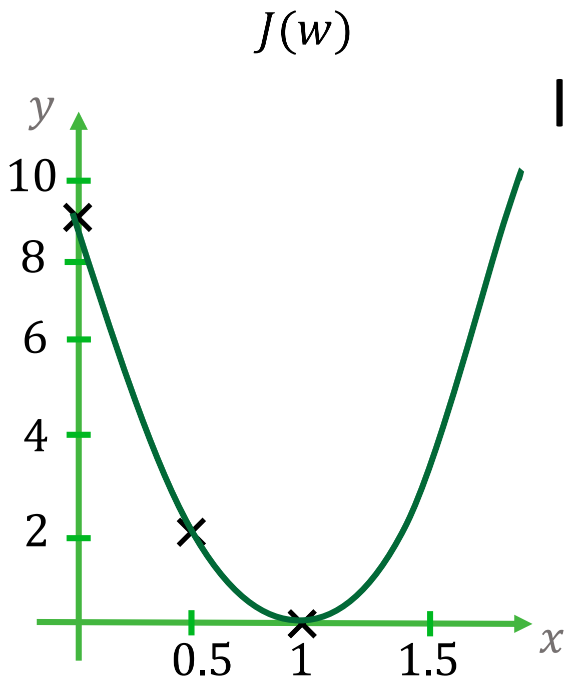
$$w = 0$$

$$J(0) = \frac{1}{2 * 3}\left((0-2)^2 + (0-4)^2 + (0-6)^2\right) = 9.3$$

$$J(0) = 9.3$$

Supervised Learning: Linear Regression

How to choose *w* to minimize *J(w)*?



In this case is when *w=1*

Supervised Learning: Linear Regression

UNIVERSIDAD DE ANTIOQUIA

1803

Model: $f_{w,b}(x) = wx + b$

Parameters: w, b

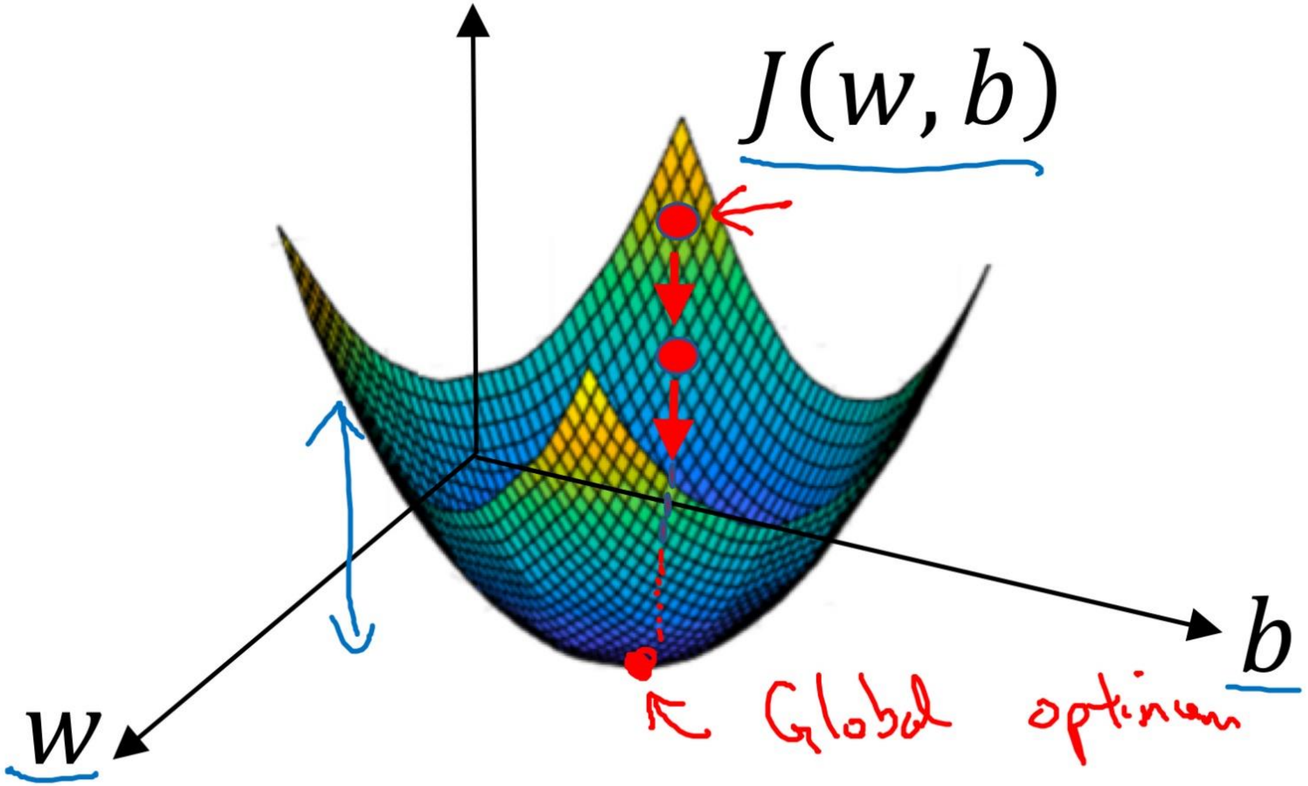
Cost function: $J(w,b) = \frac{1}{2m}\sum_{i=1}^{m}(f_{w,b}(x^{(i)}) - y^{(i)})^2$

Goal: $\underset{w,b}{minimize}\, J(w,b)$

Supervised Learning: Linear Regression

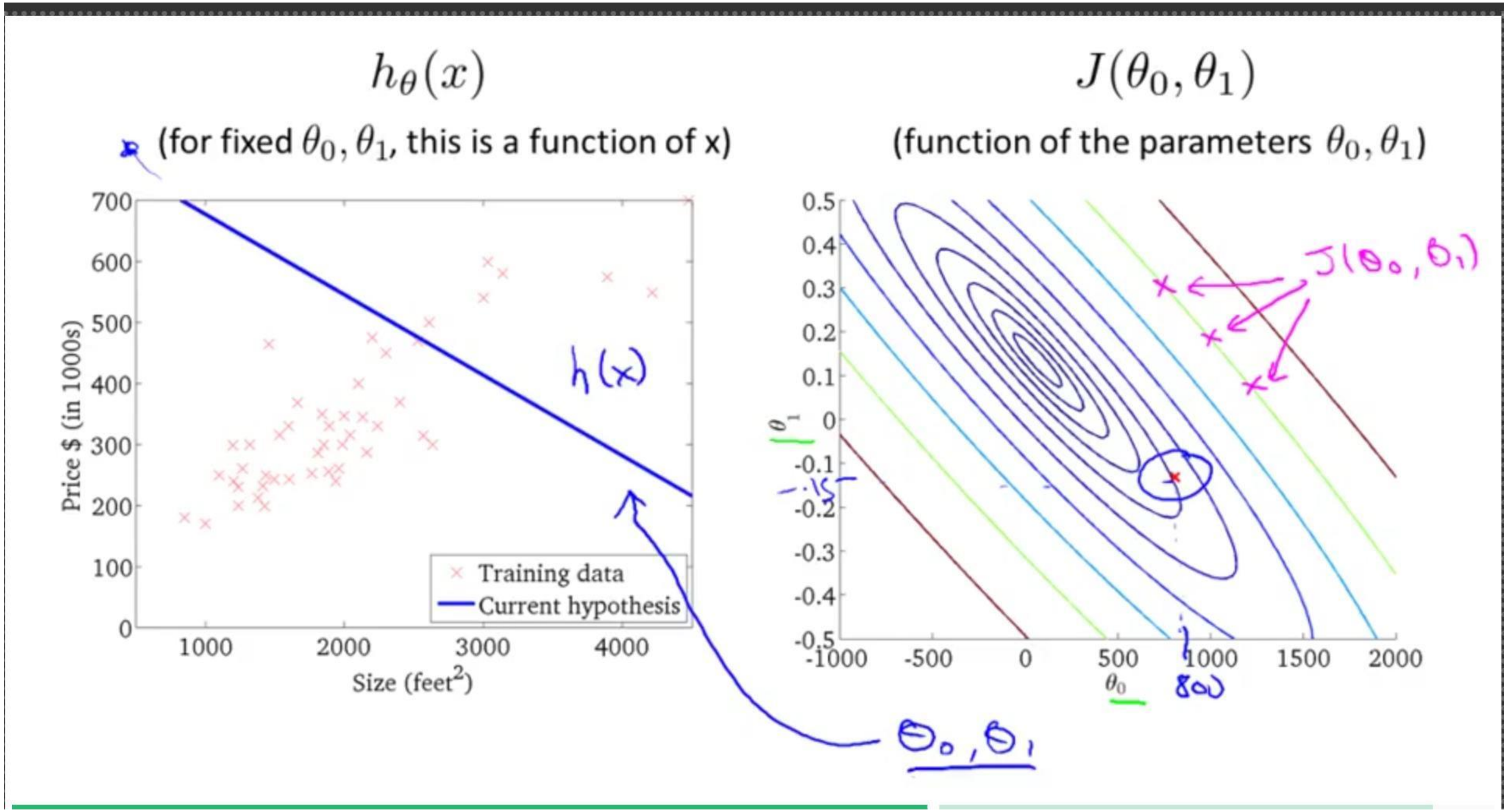
UNIVERSIDAD DE ANTIOQUIA

1803



One global minimum in Squared Error Function

Supervised Learning: Linear Regression



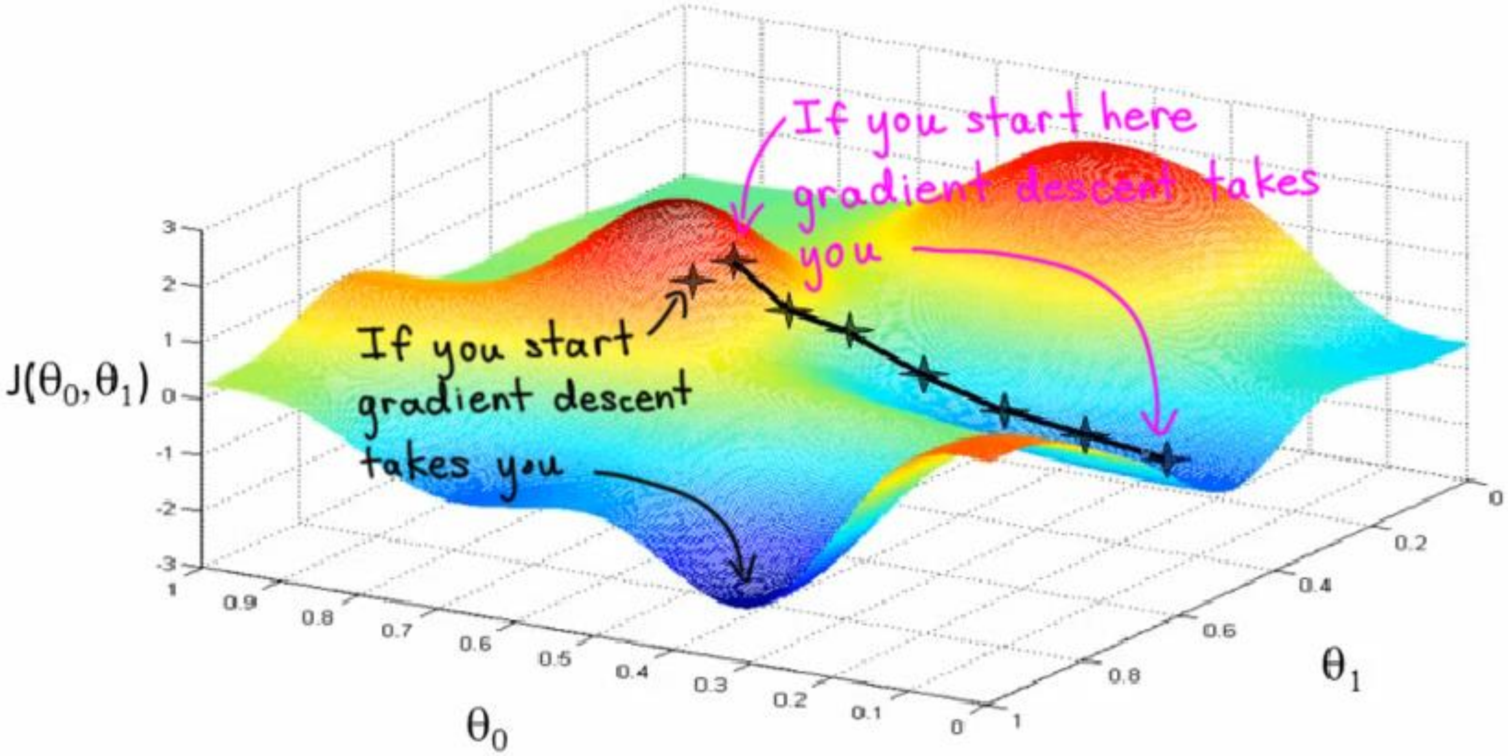
Contour plots

Gradient Descent: minimize any function

Outline:

- Start with some *w, b*
- Change *w, b* to reduce *J(w,b)*
- Keep changing until at or near a minimum

Gradient Descent: minimize any function



More tan one local Minimum: Neural Networks

Gradient Descent algorithm:

$$Repeat\ until\ convergence$$

$$w = w - \alpha \frac{\partial}{\partial w} J(w, b)$$

$$b = b - \alpha \frac{\partial}{\partial b} J(w, b)$$

α: Learning rate

Control the steps of the algorithm to update the parameters

Gradient Descent algorithm:

$$w\_temp = w - \alpha \frac{\partial}{\partial w} J(w, b)$$

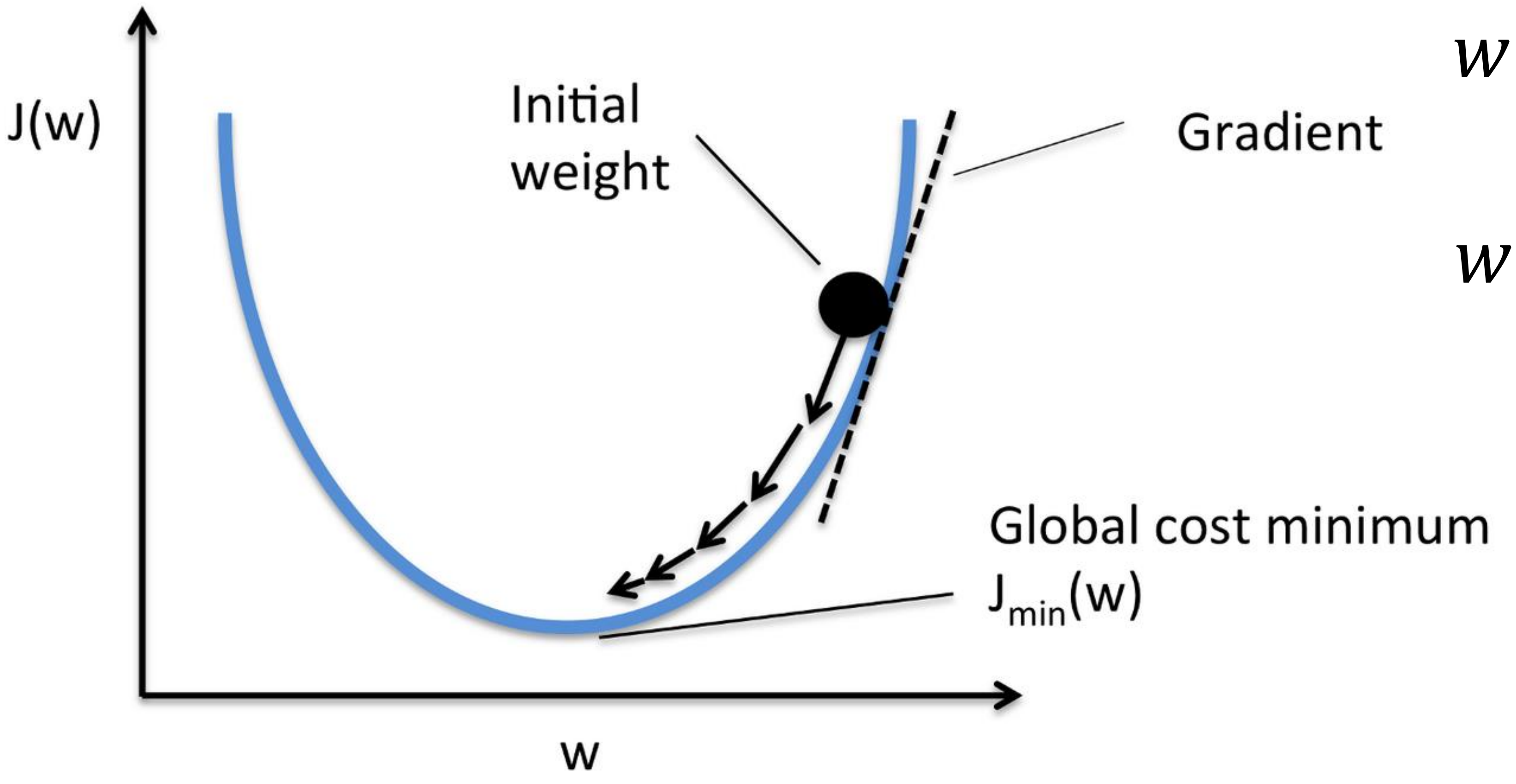
Simultaneous update

$$b\_temp = b - \alpha \frac{\partial}{\partial b} J(w, b)$$

$$w = w\_temp$$

$$b = b\_temp$$

Gradient Descent algorithm:

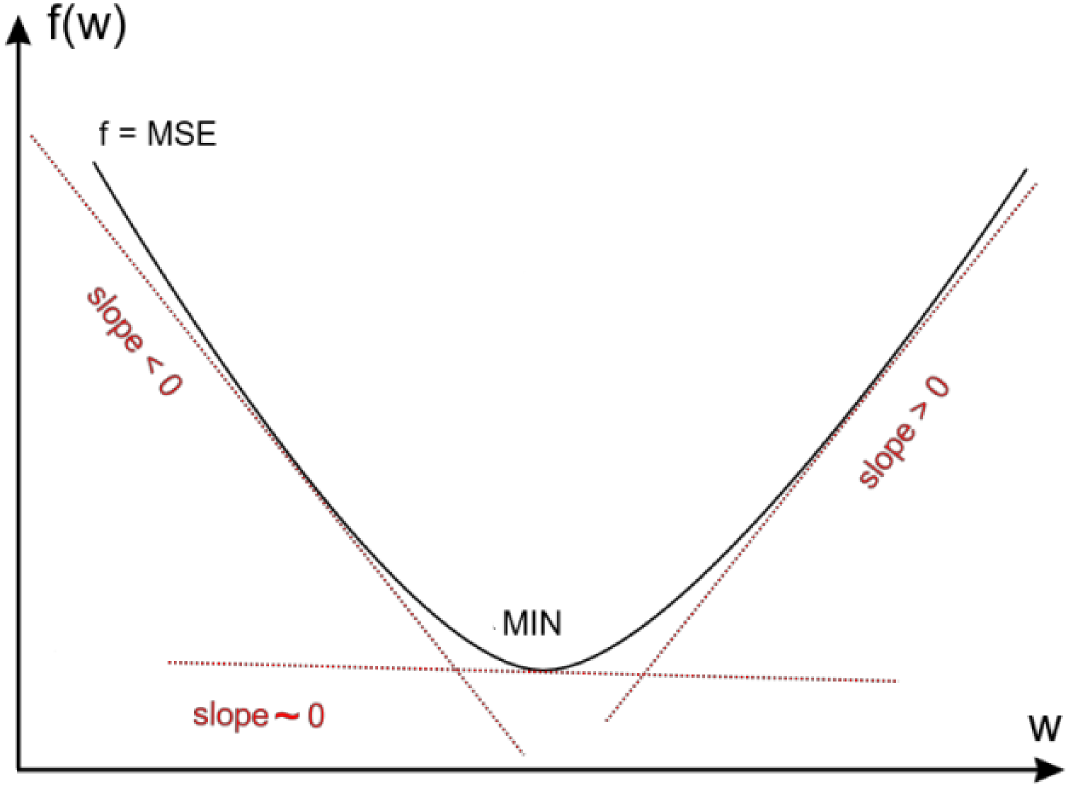


$$w = w - \alpha \frac{\partial}{\partial w} J(w, b)$$

$$w = w - \alpha * positive$$

Supervised Learning: Linear Regression

Gradient Descent algorithm:



$$w = w - \alpha \frac{\partial}{\partial w} J(w, b)$$

$$w = w - \alpha * positive$$

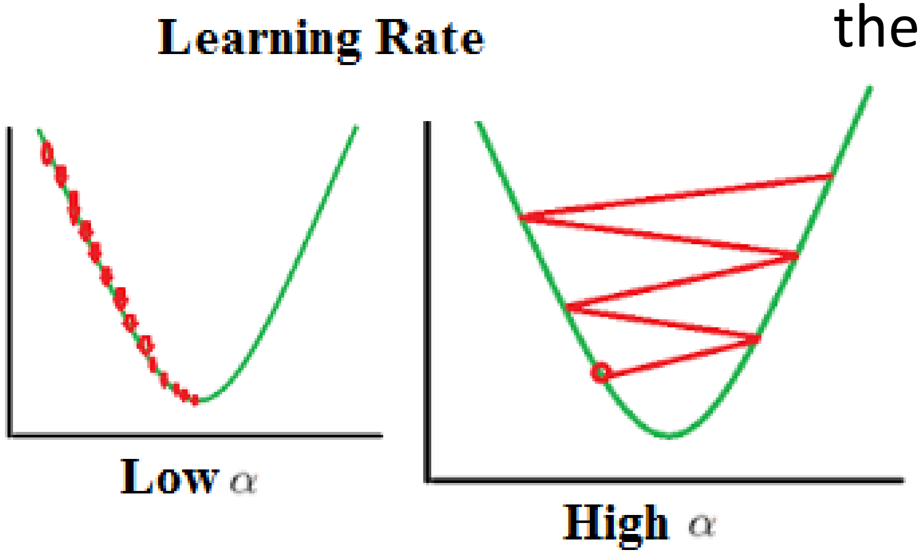
$$w = w - \alpha * negative$$

Supervised Learning: Linear Regression

Learning Rate:

If ∝ is too small, gradient descent can be very slow

If ∝ is too large, gradient descent can never reach the minimum (diverge)



When near a local minimum, gradient descent takes smaller steps (derivative smaller)

Gradient Descent for Linear Regression: $f_{w,b}(x) = wx + b$

$$J(w,b) = \frac{1}{2m}\sum_{i=1}^{m}(f_{w,b}(x^{(i)}) - y^{(i)})^2$$

$$\frac{\partial}{\partial w}J(w,b) = \frac{1}{m}\sum_{i=1}^{m}(f_{w,b}(x^{(i)}) - y^{(i)})x^{(i)}$$

$$\frac{\partial}{\partial b}J(w,b) = \frac{1}{m}\sum_{i=1}^{m}(f_{w,b}(x^{(i)}) - y^{(i)})$$

Supervised Learning: Linear Regression

UNIVERSIDAD DE ANTIOQUIA

1803

Gradient Descent Algorithm:

$$f_{w,b}(x) = wx + b$$

Repeat until convergence {

$$w = w - \alpha \frac{1}{m} \sum_{i=1}^{m} (f_{w,b}(x^{(i)}) - y^{(i)}) x^{(i)}$$

$$b = b - \alpha \frac{1}{m} \sum_{i=1}^{m} (f_{w,b}(x^{(i)}) - y^{(i)})$$

}

“Batch” Gradient Descent:

All the training examples are used in each step of the algorithm.

Linear Regression with Multiple features (variables)

Size in m^2 (x_1)	Number of bedrooms (x_2)	Number of bathrooms (x_3)	Price in millions (y)
20	1	1	60
30	1	1	100
40	1	1	120
50	1	2	180
…	…	…	…

$x_j = j^{th}\ feature$

$n = number\ of\ features$

$\vec{x}^{(i)} = features\ of\ i^{th}\ training\ example$

Linear Regression with Multiple features (variables)

Size in m^2 (x_1)	Number of bedrooms (x_2)	Number of bathrooms (x_3)	Price in millions (y)
20	1	1	60
30	1	1	100
40	1	1	120
50	1	2	180
…	…	…	…

$x_j = j^{th}\ feature$

$n = number\ of\ features$

$\vec{x}^{(i)} = features\ of\ i^{th}\ training\ example$

Linear Regression with Multiple features (variables)

Size in m^2 (x_1)	Number of bedrooms (x_2)	Number of bathrooms (x_3)	Price in millions (y)
20	1	1	60
30	1	1	100
40	1	1	120
50	1	2	180
…	…	…	…

$x_j = j^{th}\ feature$

$n = number\ of\ features$

$x^{(i)} = features\ of\ i^{th}\ training\ example$

$x_j^{(i)} = feature\ j\ in\ the\ i^{th}\ training\ example$

Linear Regression with one variable:

$$f_{w,b}(x) = wx + b$$

Linear Regression with multiple variables:

$$f_{w,b}(x) = w_1x_1 + w_2x_2 + w_3x_3 + \cdots + w_n x_n + b$$

Linear Regression with multiple variables:

$$w = [w_1\ w_2\ w_3\ w_4 \ldots w_n]$$

$$x = [x_1\ x_2\ x_3\ x_4 \ldots x_n]$$

$$f_{w,b}(x) = w \cdot x + b$$

Dot product

Multiple Linear Regression

Gradient Descent for Multiple Linear Regression:

Repeat until convergence {

$$w_1 = w_1 - \alpha \boxed{\frac{1}{m}\sum_{i=1}^{m}\left(f_{w,b}\left(x^{(i)}\right) - y^{(i)}\right)x^{(i)}} \rightarrow \frac{\partial}{\partial w_1}J(w,b)$$

....

$$w_n = w_n - \alpha \boxed{\frac{1}{m}\sum_{i=1}^{m}\left(f_{w,b}\left(x^{(i)}\right) - y^{(i)}\right)x^{(i)}} \rightarrow \frac{\partial}{\partial w_n}J(w,b)$$

$$b = b - \alpha \frac{1}{m}\sum_{i=1}^{m}\left(f_{w,b}\left(x^{(i)}\right) - y^{(i)}\right)$$

}

Gradient Descent for Multiple Linear Regression:

Repeat until convergence {

$$w_1 = w_1 - \alpha \boxed{\frac{1}{m}\sum_{i=1}^{m}\left(f_{w,b}\left(x^{(i)}\right) - y^{(i)}\right)x^{(i)}} \rightarrow \frac{\partial}{\partial w_1}J(w,b)$$

....

$$w_n = w_n - \alpha \boxed{\frac{1}{m}\sum_{i=1}^{m}\left(f_{w,b}\left(x^{(i)}\right) - y^{(i)}\right)x^{(i)}} \rightarrow \frac{\partial}{\partial w_n}J(w,b)$$

$$b = b - \alpha \frac{1}{m}\sum_{i=1}^{m}\left(f_{w,b}\left(x^{(i)}\right) - y^{(i)}\right)$$

}

UNIVERSIDAD DE ANTIOQUIA

1803

THANK YOU ANDREW NG

References

- Christopher M Bishop et al. Pattern recognition and machine learning. Springer New York, 2006.
- Ng, A. (2023). Machine Learning Specialization Coursera. Standford University, DeepLearning.AI
- Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. IBM Journal of research and development, 3(3), 210-229.

UNIVERSIDAD DE ANTIOQUIA

1803

@UdeA @universidaddeantioquia @UdeA

@universidaddeantioquia @UdeA