

***COMP 7990***  
***Principles and Practices of***  
***Data Analytics***

*Lecture 3: Data Preprocessing and Univariate Linear Regression*

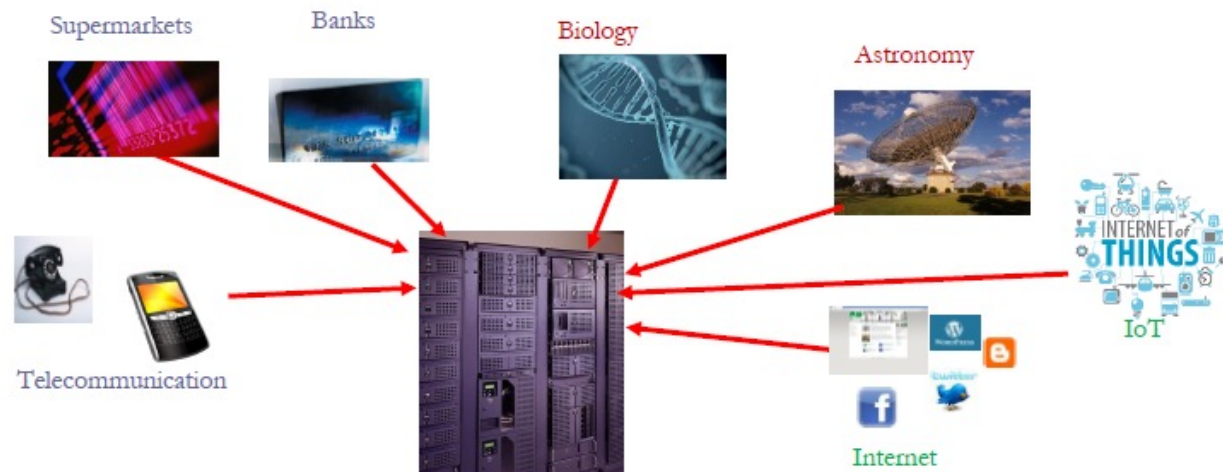
*Dr. Eric Lu Zhang*

# What is data analytics and data mining?


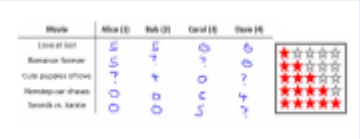


- Data Analytics: Entire process of data collection, inspecting, cleansing, transforming and modeling, interpretation and reporting.
- Data mining (knowledge discovery from data)
  - Extract of interesting (non-trivial, implicit, previously unknown potential useful) pattern or knowledge from huge amount of data.
- Data analytics is a more general concept than data mining.

# Why we need data mining?

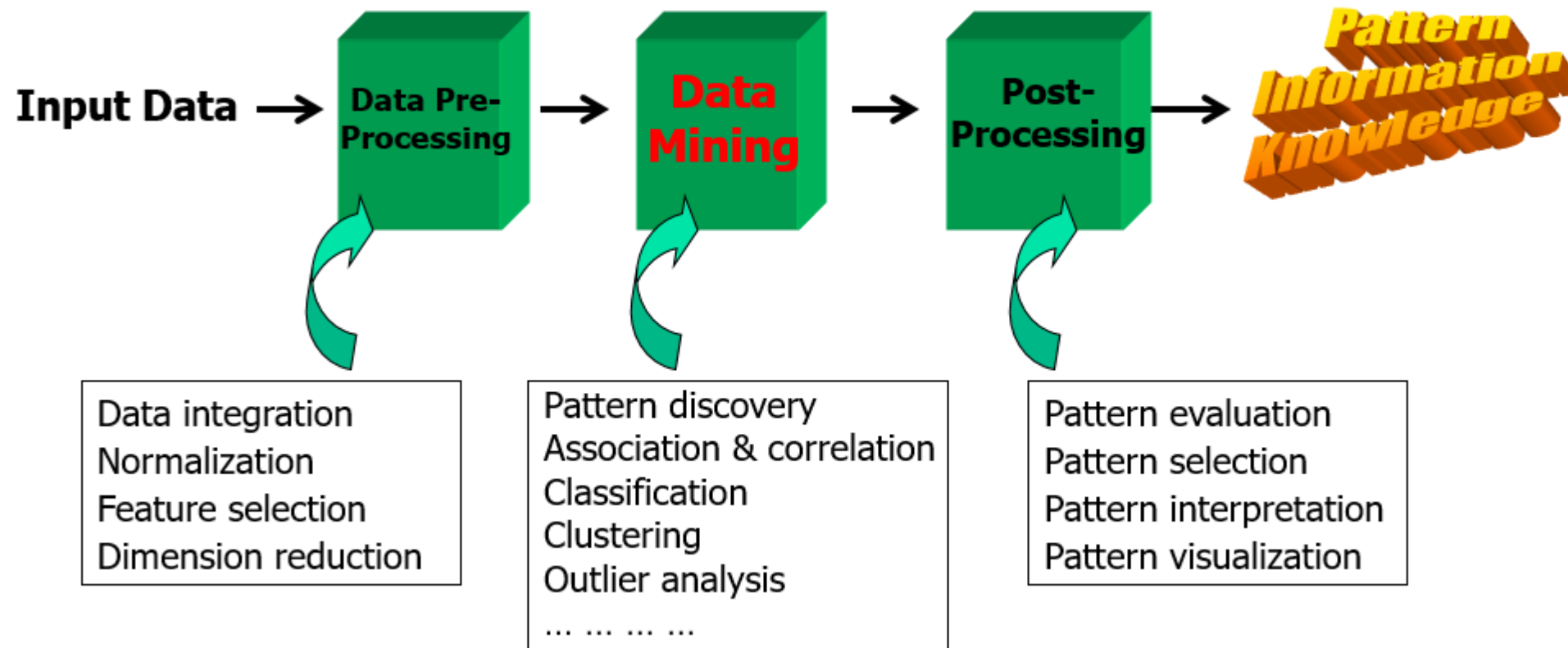
- Huge amounts of data are collected from different domains
- “We are drowning in information but starving for knowledge”-John Naibett
- The amount and the complexity of the collected data does not allow for manual analysis: we need automated analysis of massive data.



# Real-life Applications of Using Data Mining

Input Data	Methods	Output
 <p>images</p>	Classification	Is it a banana (or an apple)?
 <p>Movie ratings</p>	Recommendation System	Recommend which movies to which users?
 <p>News articles</p>	Clustering	What are the topics people discussed about in the news today?
 <p>English and Chinese sentences</p>	Classification	Translation

# Knowledge Discovery in Database (KDD): From machine learning perspective



# Outline for Data Preprocessing and Data Mining

- **Data Preprocessing**
- **Supervised learning**

- ❖ Regression

1. Linear regression with one variable
2. Linear Regression with multiple variables
3. The relationship between Correlation and Regression

- ❖ Classification

1. Perceptron
2. Artificial Neural Network
3. Support Vector Machine
4. K Nearest Neighbor

- **Unsupervised learning**

1. K-means Clustering
2. Hierarchical Clustering

# Outline for Data Preprocessing and Data Mining

- **Data Preprocessing**

- **Supervised learning**

  - ❖ Regression

1. Linear regression with one variable
2. Linear Regression with multiple variables
3. The relationship between Correlation and Regression

  - ❖ Classification

1. Perceptron
2. Artificial Neural Network
3. Support Vector Machine
4. K Nearest Neighbor

- **Unsupervised learning**

1. K-means Clustering
2. Hierarchical Clustering

# Outline for Data Preprocessing

- Why we need data preprocessing
- What is data preprocessing
- How to do data preprocessing
  - Data Clean
  - Data Transformation



# Why we need preprocess the data?

- Real world data is dirty
  - Incomplete: lacking attribute values
    - e.g., Occupation = " "
  - Noisy: containing errors or outliers
    - E.g., Salary = "-10"
  - Inconsistent: containing discrepancies in codes or names
    - E.g., Age = "40", Birthday = '03/02/1990'
    - E.g., Was rating "1, 2, 3", now rating "A, B, C"
    - E.g, discrepancy between duplicate records

ID	Birthday	Age	Salary	Occupation
1001	11/01/1986	32	-10	Engineer
1002	03/02/1990	40	30k	Manager
1003	01/01/1980	39	40k	

Noisy value

Inconsistent value

Missing value

# Why Data is Dirty?

- Incomplete data may come from
  - “Not applicable” data value
    - E.g., annual income is not applicable to children
  - People do not want to disclose the information
    - E.g., age, birthday
  - Human/hardware/software problems
    - E.g., data was accidentally deleted
- Noisy Data may come from
  - Faulty data collection instruments
  - Human/Computer errors
- Inconsistent data may come from
  - Different data sources

# Why Data Preprocessing is important

- No quality data, no quality results
  - Quality decisions must be based on quality data
    - E.g., missing data or incorrect data may cause incorrect or even misleading statistics
- **Garbage In, Garbage Out**
- In general, data pre-processing consumes more than 60% of a data analytics project effort.

# Outline

- Why we need data preprocessing
- What is data preprocessing
- How to data preprocessing
  - Data Clean
  - Data Transformation

# Typical tasks in data preprocessing

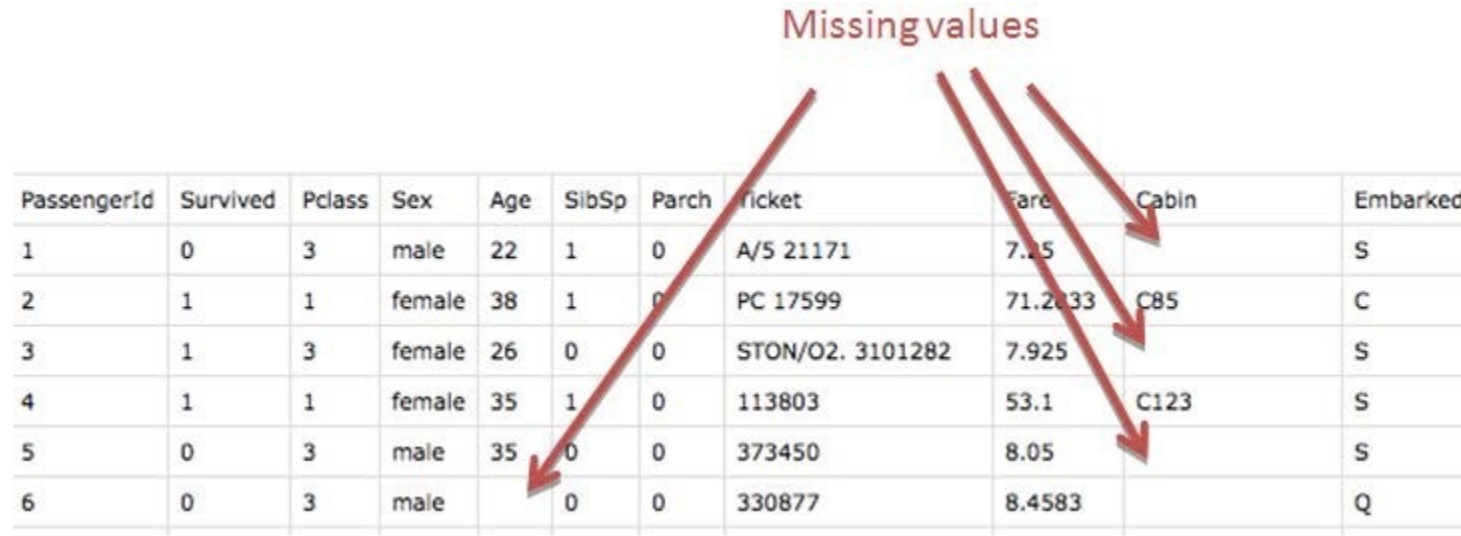
- Data Cleaning
  - Fill in missing values
  - Smooth noisy data
  - Identify and remove outliers
- Data transformation and data discretization
  - Feature type conversion
  - Normalization
    - Scaling attribute values to fall within a specific range (e.g., [0, 1])
  - Attribute construction

# Data Cleaning: Handling Incomplete (Missing) Values

- Data is not always available
  - E.g, many data samples do not have recorded value for several attributes, such as customer age, customer income in sales data
- Missing data may be due to
  - Equipment malfunction
  - Data is not entered
  - Certain data may not be considered at the time of data collection

# Missing Data Example (Titanic Data)

- Titanic Data



Missing values

PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	male	22	1	0	A/5 21171	7.25		S
2	1	1	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	female	26	0	0	STON/O2. 3101282	7.925		S
4	1	1	female	35	1	0	113803	53.1	C123	S
5	0	3	male	35	0	0	373450	8.05		S
6	0	3	male		0	0	330877	8.4583		Q

Description for each feature contained in this dataset:

- Survival: Survival 0 = No, 1 = Yes
- Pclass: A proxy for economic status (1 = 1<sup>st</sup> class, 2 = 2<sup>nd</sup> class, 3 = 3<sup>rd</sup> class)
- SibSp: number of siblings / spouses aboard the Titanic
- Parch: number of parents / children aboard the Titanic
- Ticket: Ticket number
- Fare: Passenger fare
- Cabin: Cabin number
- embarked: Port of Embarkation C = Cherbourg, Q = Queenstown, S = Southampton

# Outline

- Why we need data preprocessing
- What is data preprocessing
- How to do data preprocessing
  - Data Clean
  - Data Transformation



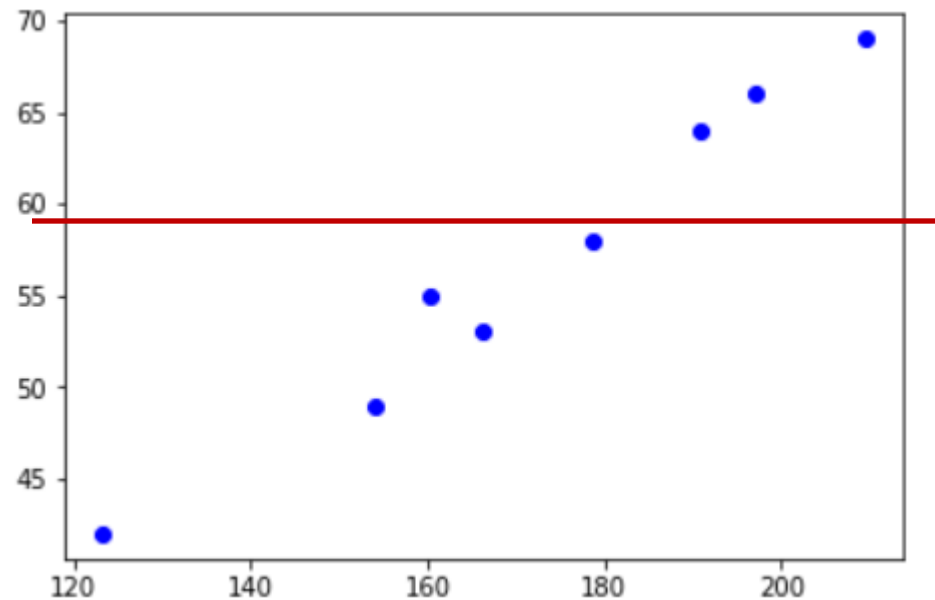
# How to handle missing data?

- Ignore the data sample with missing values
  - Not a good solution, especially when data is scarce
- Ignore attributes with missing values
  - Use only attributes (features) with all values
  - May leave out important features
- Fill in it by
  - A global constant: e.g., “unknown”
  - Attribute mean/Median/mode
  - Predict the missing value (data imputation)
    - Estimate gender based on first name (name gender)
    - Estimate age based on first name (name popularity)
    - Build a predictive model based on other attributes/features

# Example of handling missing value by mean

height	weight
123.20	42.0
138.60	NaN
154.00	49.0
160.16	55.0
166.32	53.0
178.64	58.0
187.88	NaN
190.96	64.0
197.12	66.0
209.44	69.0

Mean value of  
weight

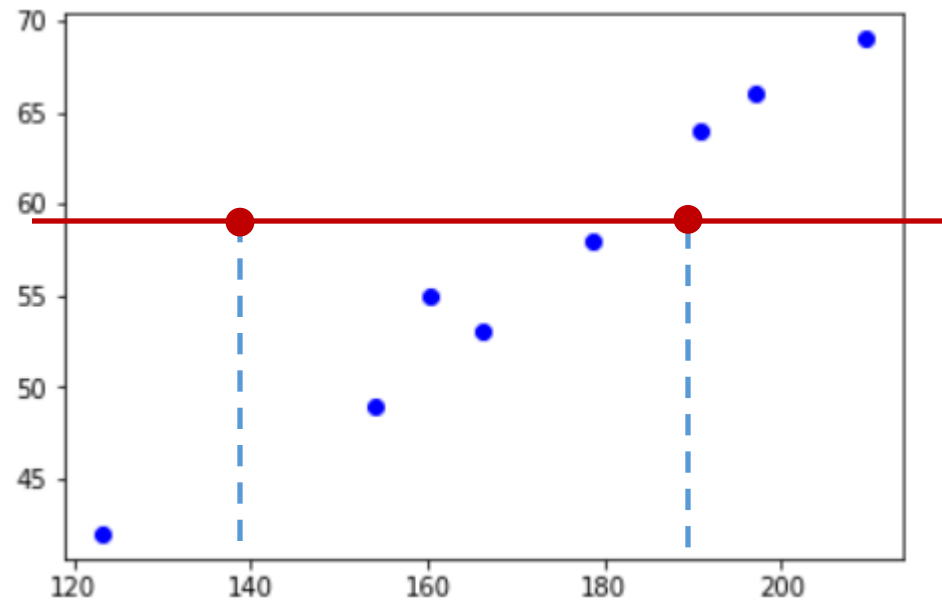


# Example of handling missing value by mean

height	weight
123.20	42.0
138.60	NaN
154.00	49.0
160.16	55.0
166.32	53.0
178.64	58.0
187.88	NaN
190.96	64.0
197.12	66.0
209.44	69.0

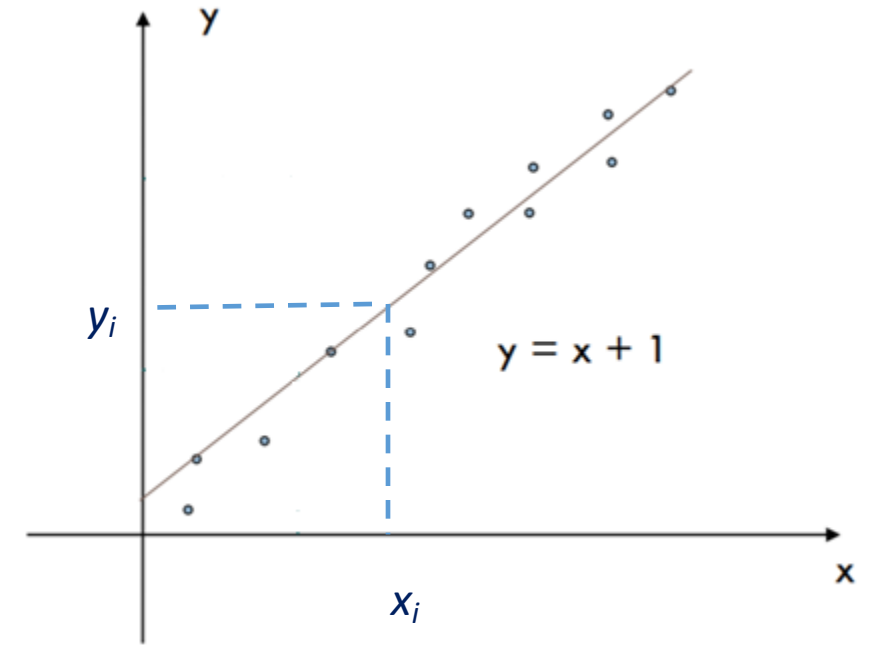
57.0

Mean value of  
weight



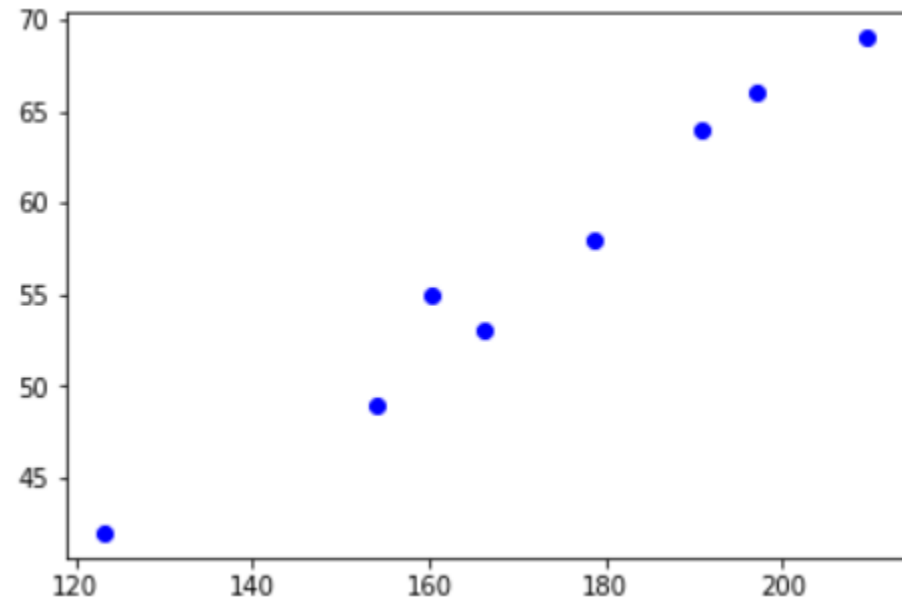
# Handling missing value by prediction model

- Replace missing value by predicted values by a prediction model (e.g., a regression model)
- Requires attribute dependencies
- Can be used for handling missing data and noisy data.
- Regression models will be discussed in deep later.



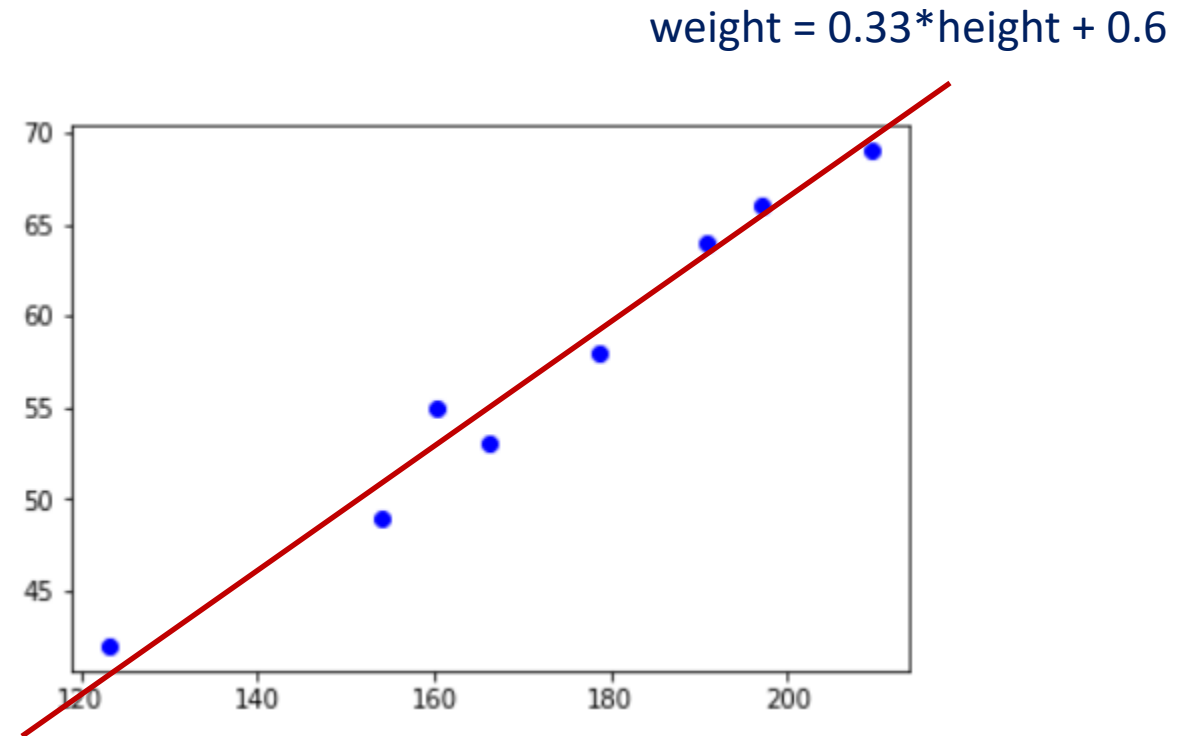
# Example of handling missing value by regression

height	weight
123.20	42.0
138.60	NaN
154.00	49.0
160.16	55.0
166.32	53.0
178.64	58.0
187.88	NaN
190.96	64.0
197.12	66.0
209.44	69.0



# Example of handling missing value by regression

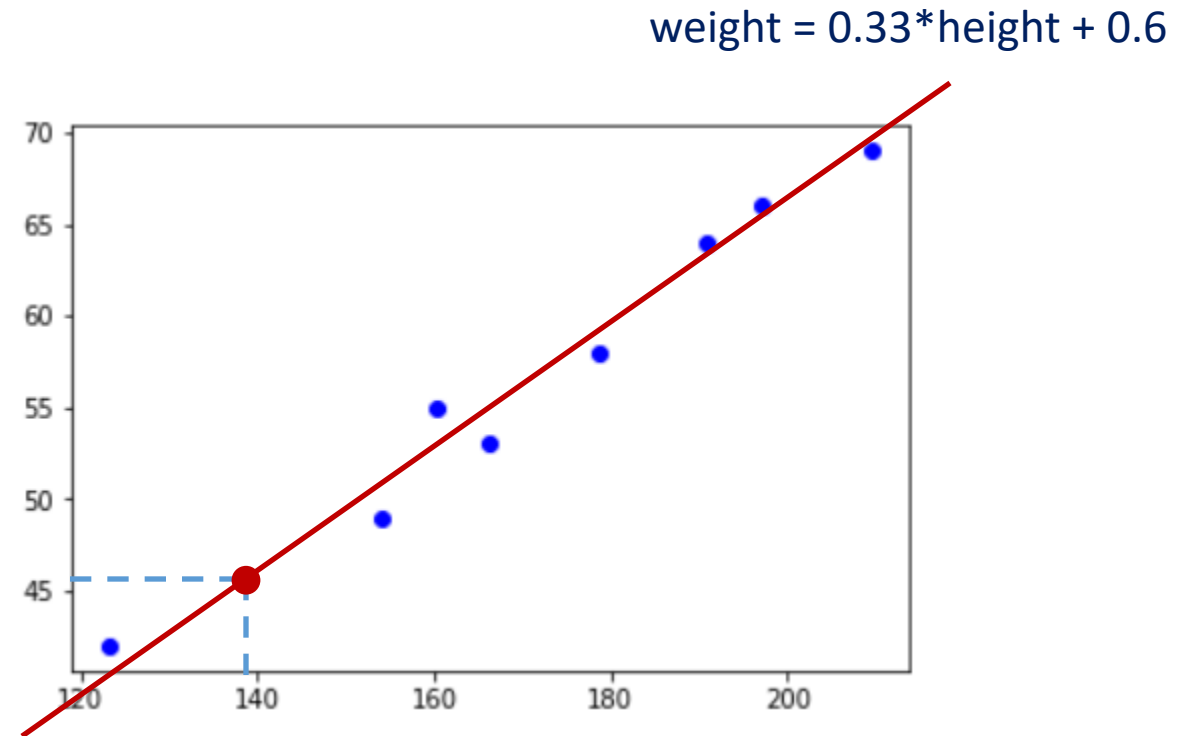
height	weight
123.20	42.0
138.60	NaN
154.00	49.0
160.16	55.0
166.32	53.0
178.64	58.0
187.88	NaN
190.96	64.0
197.12	66.0
209.44	69.0



# Example of handling missing value by regression

height	weight
123.20	42.0
138.60	NaN
154.00	49.0
160.16	55.0
166.32	53.0
178.64	58.0
187.88	NaN
190.96	64.0
197.12	66.0
209.44	69.0

46.34

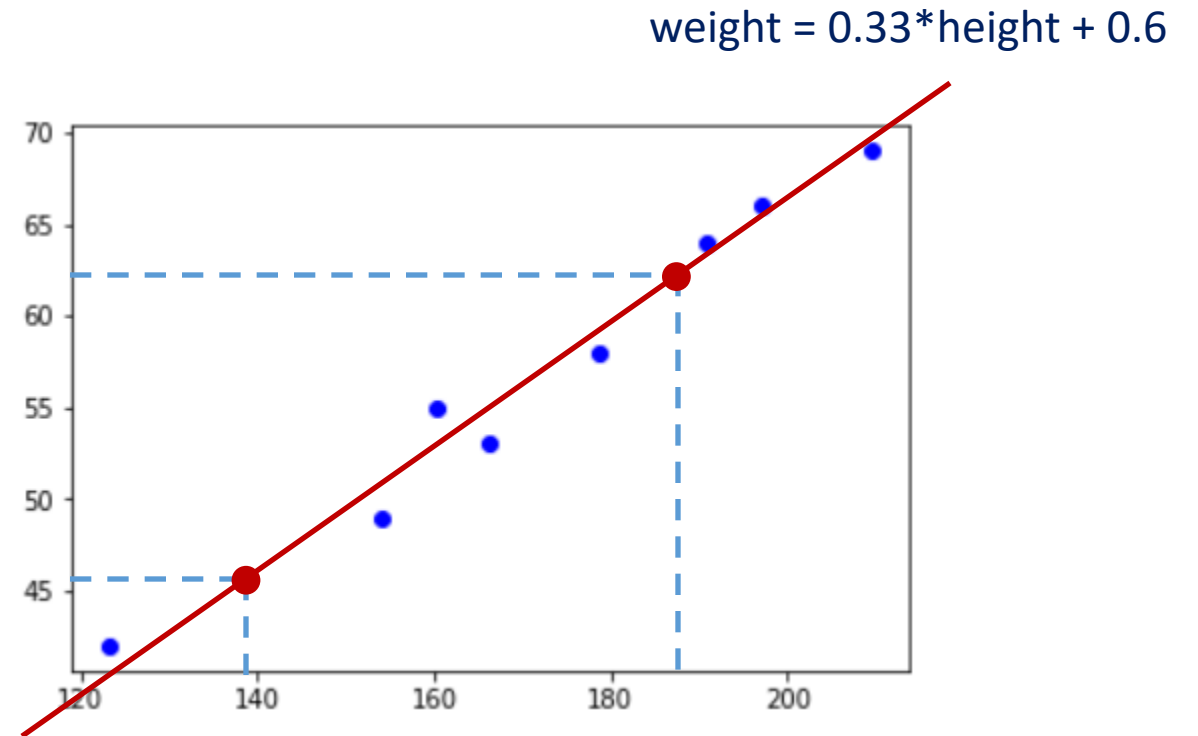


# Example of handling missing value by regression

height	weight
123.20	42.0
138.60	NaN
154.00	49.0
160.16	55.0
166.32	53.0
178.64	58.0
187.88	NaN
190.96	64.0
197.12	66.0
209.44	69.0

46.34

62.60





# Data Cleaning: Handling Noisy Data

- Noise: random error or variance in a measured variable
- Incorrect attribute values may be due to
  - Errors in data collection devices
  - Wrong input
  - Technology limitation

# How to Handle Noisy Data

- Binning
  - First sort data and partition into bins
  - Smooth by bin mean/median/boundaries
- Regression
  - Smooth by fitting the data into regression functions
- Clustering
  - Detect and remove outliers

# Simple Discretization Methods: Binning

- Equal-width (distance) Partitioning
  - Divides the range into  $N$  intervals of equal size: uniform grid
  - Suppose  $min$  and  $max$  are the lowest and highest values of the attribute, the width of intervals should be:  $w = (max - min)/N$
  - The most straight-forward method
  - Outliers may dominate presentation
  - Skewed data is not handled well
- Equal-depth (frequency) partitioning
  - Divides the range into  $N$  intervals, each containing approximately same the number of samples
  - Skewed data is also handled well

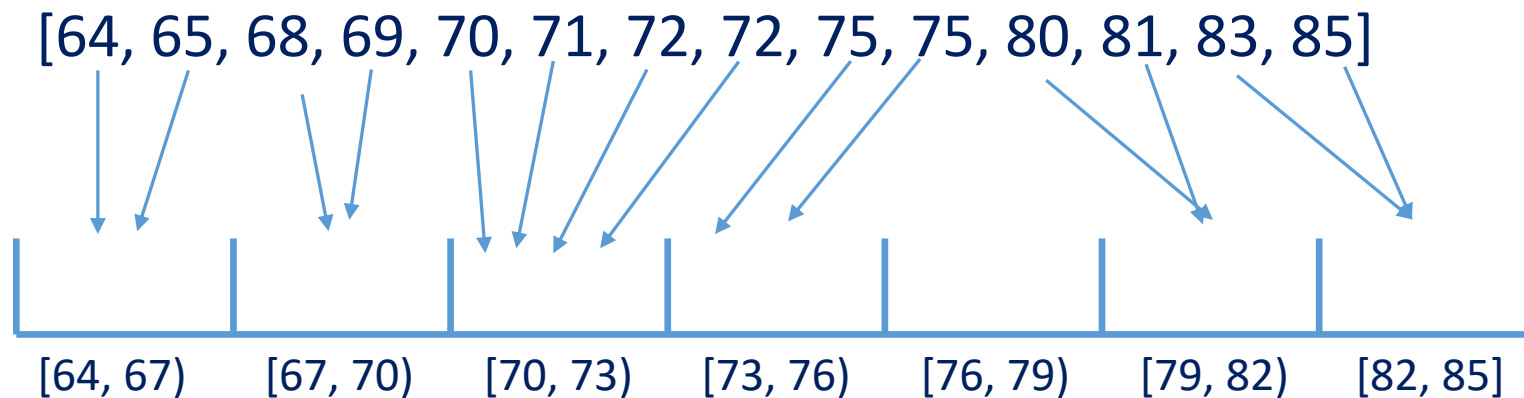
# Example of Equal-width Binning for data smoothing

- Suppose we have the following values for temperature (data should be sorted first) and we want to divided them into 7 bins

[64, 65, 68, 69, 70, 71, 72, 72, 75, 75, 80, 81, 83, 85]

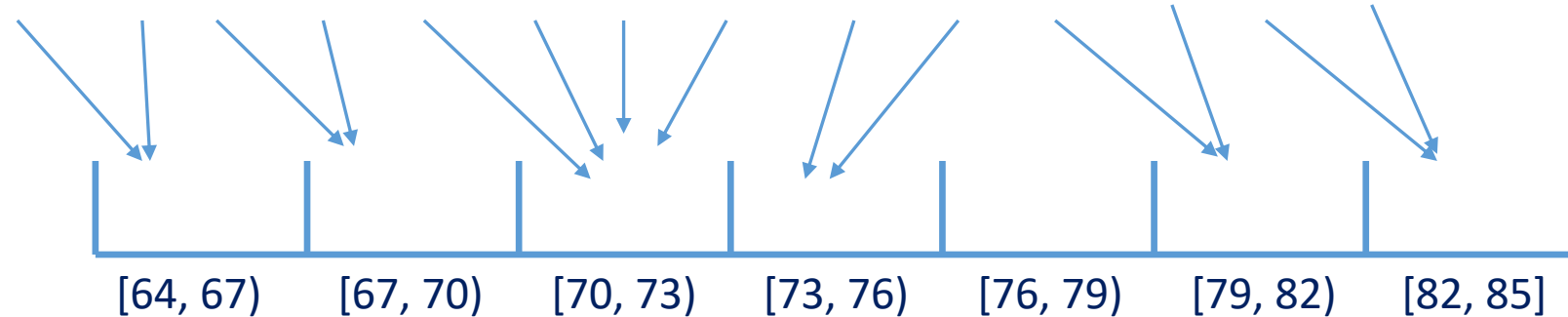
- Partition data into bins

- Compute the width  $w = (85-64)/7 = 3$



# Example of Equal-width Binning for data smoothing

[64, 65, 68, 69, 70, 71, 72, 72, 75, 75, 80, 81, 83, 85]



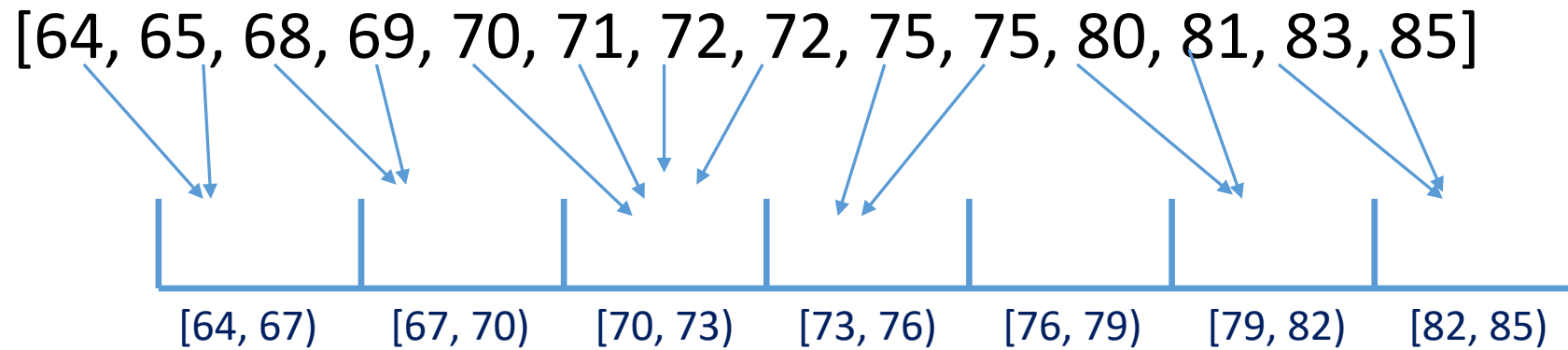
- Smoothing by bin means

- Each value in a bin is replaced by the mean value of the bin

[64.5, 64.5, 68.5, 68.5, 71.25, 71.25, 71.25, 71.25, 75, 75, 80.5, 80.5, 84, 84]

- Similarly, smoothing by bin medians can be used, in which each bin value is replaced by the bin median.

# Example of Equal-width Binning for data smoothing



- Smoothing by bin boundaries
  - Bin boundaries are the minimum and maximum values in a given bin.
  - Each bin value then is replaced by the closest boundary value

[64, 65, 68, 69, 70, 70, 72, 72, 75, 75, 80, 81, 83, 85]

In general, the larger the width, the greater the effect of the smoothing

# Example of Equal-width Binning

- Advantage
  - Simple and easy to implement
  - Produce a reasonable abstraction of data
- Disadvantage
  - Where does  $N$  come from?
  - Sensitive to outliers

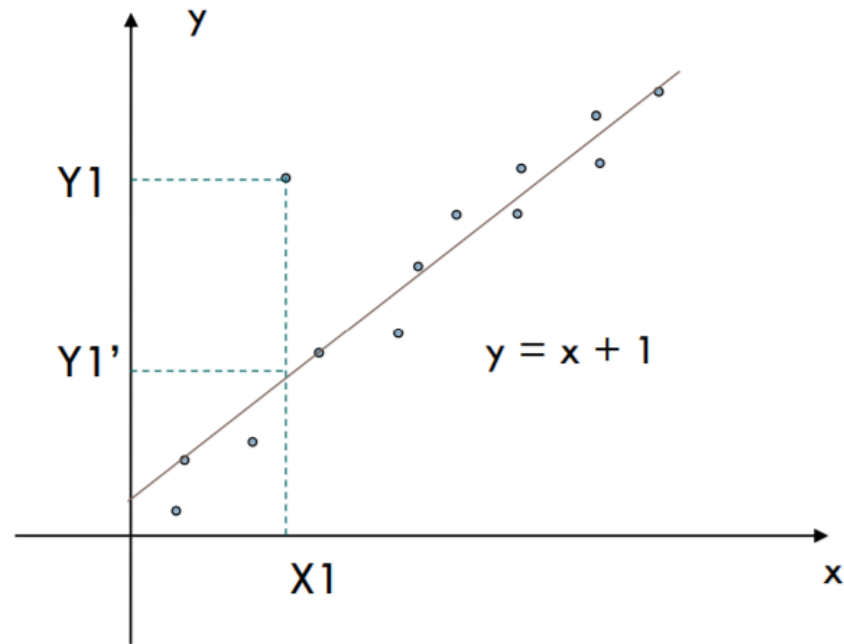
# Example of Equal-depth Binning

- Divides the range into  $N$  intervals, each containing **approximately the same number of samples**
- E.g., we have the following values for prices and we want to divided them into 3 bins using Equal-depth binning  
[4, 8, 15, 21, 21, 24, 25, 28, 34]
- Partition into 3 bins (equal frequency)  
[4, 8, 15, 21, 21, 24, 25, 28, 34]
- Smooth by bin means  
[9, 9, 9, 22, 22, 22, 29, 29, 29]
- Smooth by bin boundaries  
[4, 4, 15, 21, 21, 24, 25, 25, 34]

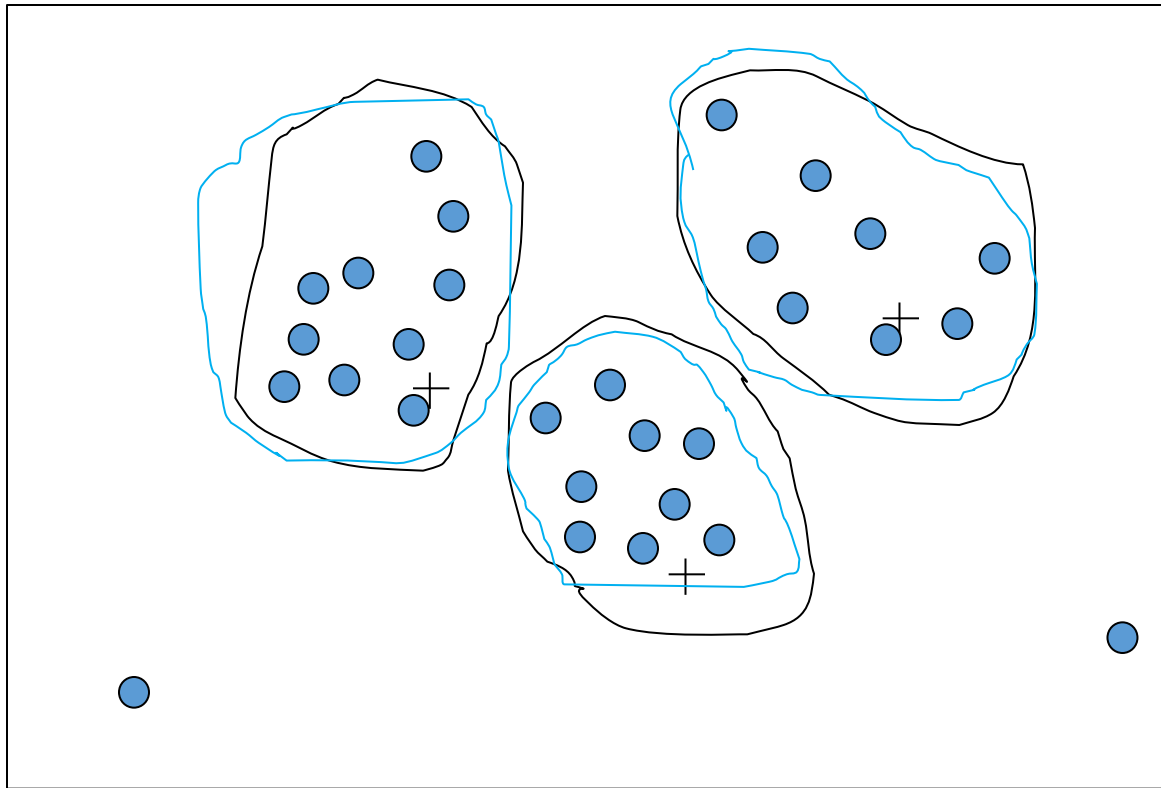


# Handling Data Noisy by Regression Analysis

- Data smoothing can also be done by regression analysis.



# Handling Data Noisy by Clustering Analysis



- Outliers may be detected by clustering analysis.
- Outliers may need to be removed from the data.
- Clustering algorithms will be discussed in depth in future lectures.

# Data Transformation

- Data Transformation
  - A function that maps the entire set of values of a given attribute to a new set of replacement values s.t. each old value can be identified with one of the new values
- Methods
  - Feature Type Conversion
  - Normalization
  - Feature construction

# Feature Type Conversion

- Some tools can only deal with nominal values but other only deal with numeric values.
- Features have to be converted to satisfy the requirements of different tools
  - Numeric -> Nominal
    - Binning
  - Nominal -> numeric
    - One hot encoding
  - Ordinal -> numeric (order matters)
    - A -> 4.0
    - A- -> 3.7
    - B+ -> 3.3
    - B -> 3

# Nominal to Numeric (one-hot encoding)

ID	Color
1	Red
2	Green
3	Blue



ID	Color_red	Color_green	Color_Blue
1	1	0	0
2	0	1	0
3	0	0	1

- One of the ways to encode the nominal variable to numeric is **one-hot encoding**
- With one-hot encoding, a nominal feature becomes a vector whose size is the number of possible choices for that features

# Normalization: motivation

- Data have attribute values
- Can we compare these attribute values?
- E.g., considering the following two records, which one is more similar to (5.9 ft, 50kg)
  - (4.6 ft, 55 kg)
  - (5.6 ft, 56 kg)
- We need to normalize data to makes different attributes comparable.

# Normalization

- For distance-based methods, normalization helps to prevent that attributes with large ranges out-weight attributes with small ranges
- Scale the attribute values to a small specified range
- Normalization Methods
  - Normalization by decimal scaling
  - Min-Max normalization (normalized by range)
  - Z-score normalization

# Normalization: Decimal Scaling

- Decimal Scaling
  - The values of an attribute are normalized by moving the decimal point.
  - The number of decimal points moved depends on the maximum absolute value of the attribute.
  - Decimal scaling maps a value  $x_i$  to  $x_i'$  by

$$x_i' = \frac{x_i}{10^j} \longrightarrow j \text{ is the smallest integer such that } \max(|x_i|) < 1$$

- E.g., suppose that the recorded values of an attribute range from -986 to 917. The maximum absolute value is 986. To normalize by decimal scaling, we therefore divide each value by 1,000 (i.e.,  $j = 3$ ) so that -986 normalizes to -0.986 and 917 normalizes to 0.917.



# Normalization: min-max Normalization

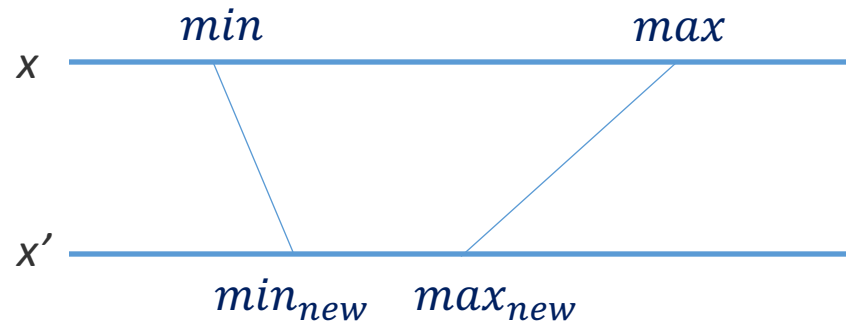
- Min-max normalization
  - Performs a linear transformation on the original data.
  - Suppose  $min$ ,  $max$  are the minimum and maximum values of an attribute and we want to normalize the attribute value to  $[min_{new}, max_{new}]$ , min-max normalization maps a value  $x_i$  to  $x_i'$  by

$$x_i' = \frac{(x_i - min)}{max - min} (max_{new} - min_{new}) + min_{new}$$

- E.g., suppose that the minimum and maximum values for the feature income are \$12,000 and \$98,000. We would like to map income to the range [0.0, 1.0]. By min-max normalization, what is the mapped value for \$73,600?

$$\frac{(73,600 - 12,000)}{98,000 - 12,000} (1.0 - 0.0) + 0.0 = 0.716$$

# Deriving the formula for min-max Normalization



Find a linear transform  
 $x' = a*x + b$

- We know  $min$  is mapped to  $min_{new}$  and  $max$  is mapped to  $max_{new}$

- $min_{new} = a*min + b$

- $max_{new} = a*max + b$

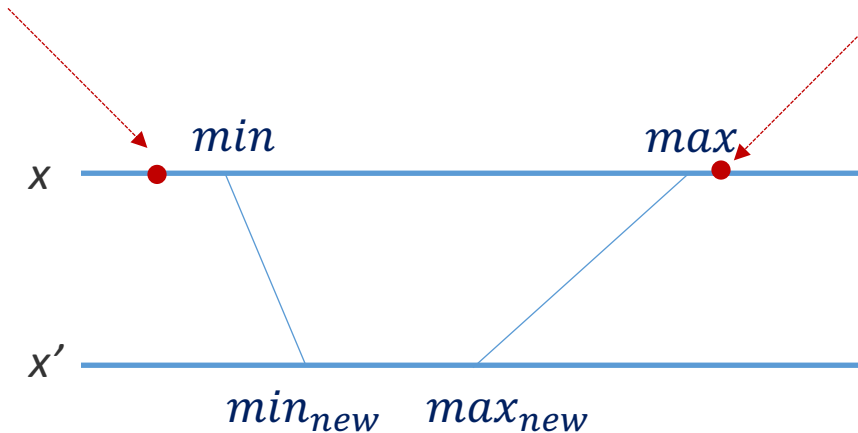
- Therefore,  $a = (max_{new} - min_{new})/(max - min);$

$$b = min_{new} - (max_{new} - min_{new}) * min / (max - min);$$

$$x'_i = \frac{(x_i - min)}{max - min} (max_{new} - min_{new}) + min_{new}$$

We take it from here

# Min-max normalization problem



- Min-max normalization will encounter an “out-of-bounds” error if a future input value is fall outside of the original data range.
- In some cases, we may do not know the minimum and maximum values of an attribute.

# Normalization: Z-score normalization

- Z-score normalization
  - The values of an attribute are normalized to a scale with a mean value of 0 and standard deviation of 1.
  - Z-score normalization maps a value  $x_i$  to  $x_i'$  by

$$x_i' = \frac{(x_i - \bar{x})}{S_x}$$

mean of an original x

standard deviation of original x

- E.g., suppose that the mean and standard deviation for income are \$54,000 and \$16,000. What is the mapped value for \$73,600?

$$\frac{(73,600 - 54,000)}{16,000} = 1.225$$

# Outline for Data Preprocessing and Data Mining

- **Data Preprocessing**
- **Supervised learning**

- ❖ Regression

1. Linear regression with one variable
2. Linear Regression with multiple variables
3. The relationship between Correlation and Regression

- ❖ Classification

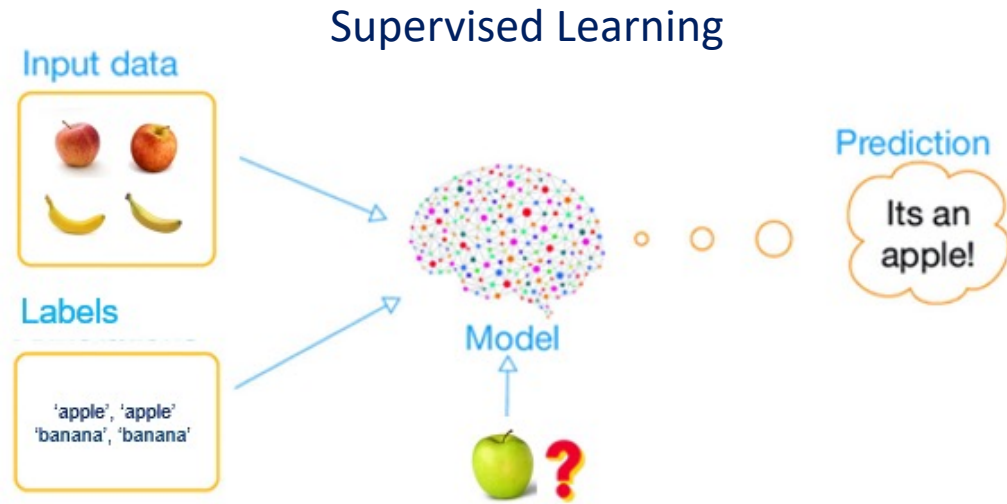
1. Perceptron
2. Artificial Neural Network
3. Support Vector Machine
4. K Nearest Neighbor

- **Unsupervised learning**

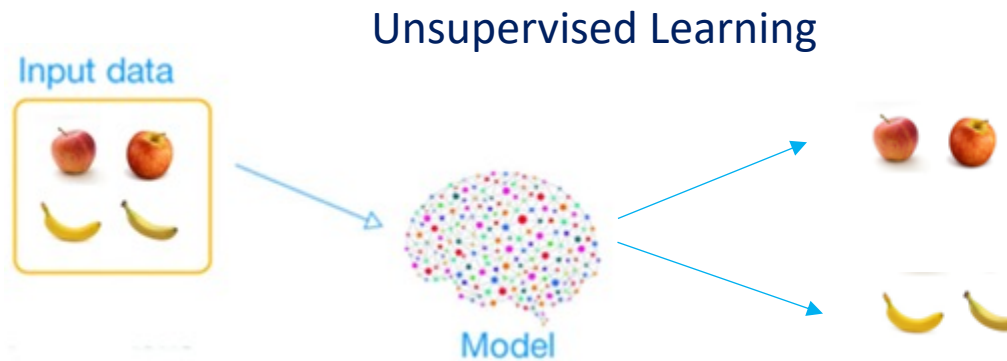
1. K-means Clustering
2. Hierarchical Clustering

# Different Types of Learning Tasks

- Supervised Learning
  - Data with labels

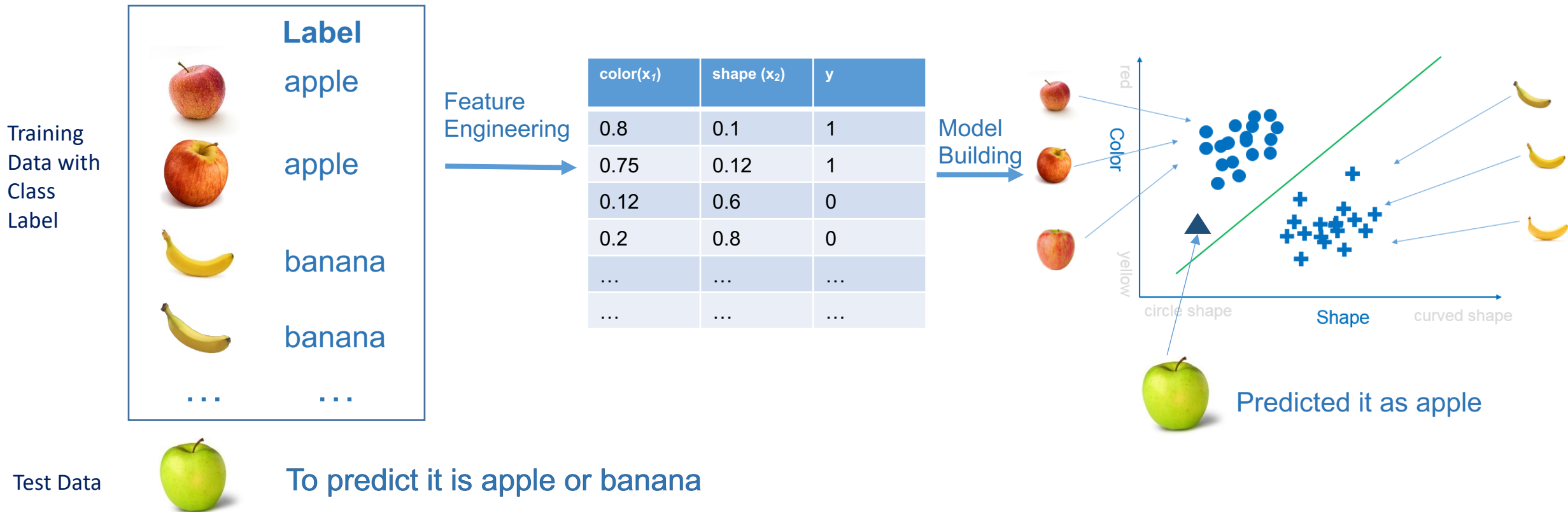


- Unsupervised Learning
  - Data without labels

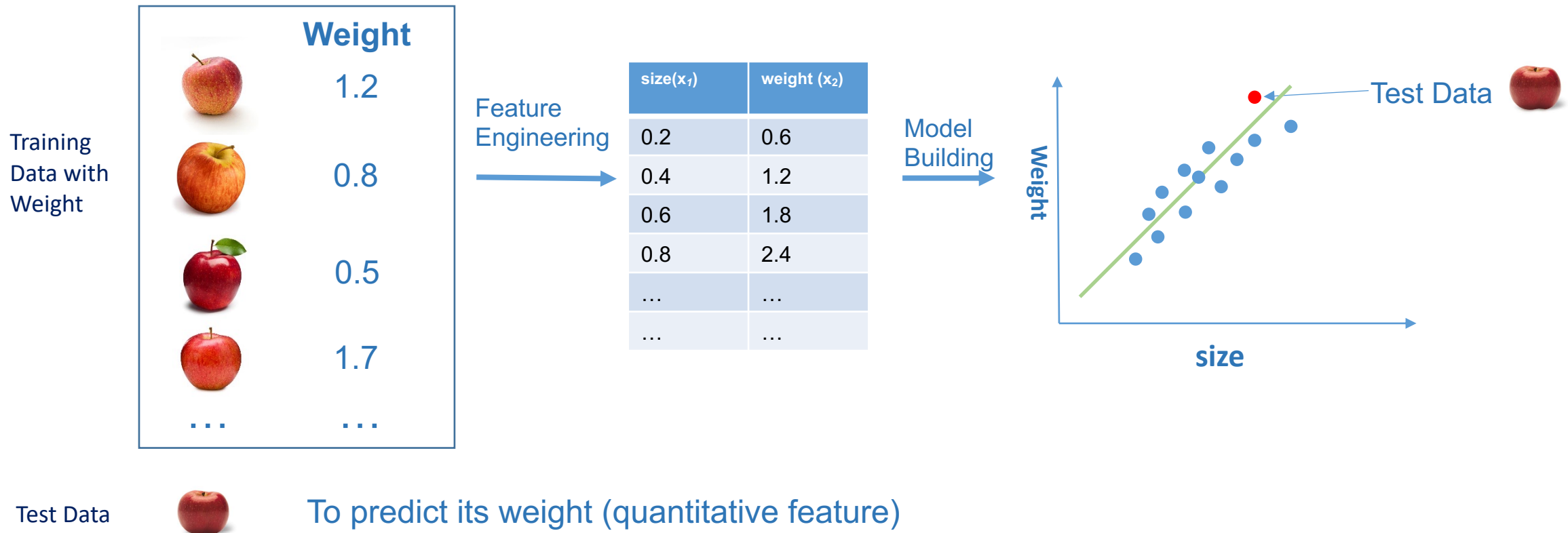


# An Example of Classification Problem

- Learn to recognize apple or banana



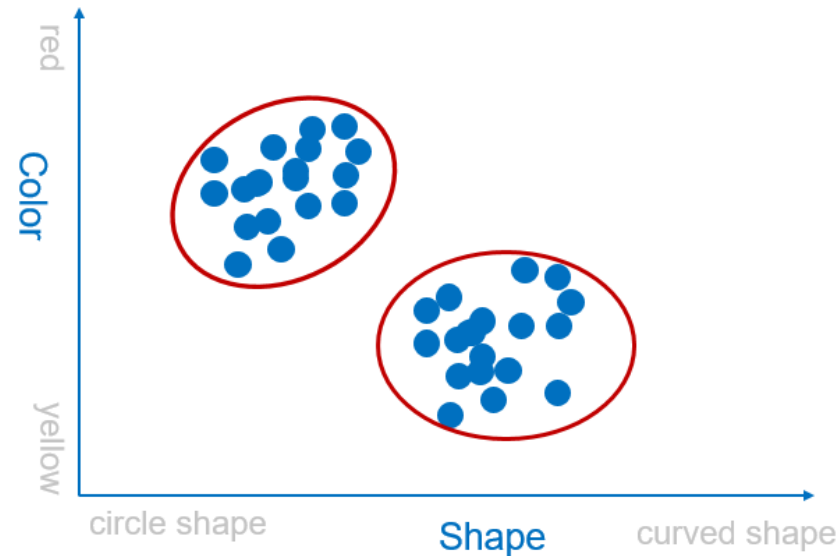
# An Example of Regression Problem





# An example for Clustering

- Clustering
  - Only the feature representation of instance is available.
  - No Label information
  - The goal is to discover groups of similar instance from data



# Outline for Data Preprocessing and Data Mining

- **Data Preprocessing**

- **Supervised learning**

- ❖ Regression

1. Linear regression with one variable
2. Linear Regression with multiple variables
3. The relationship between Correlation and Regression

- ❖ Classification

1. Perceptron
2. Artificial Neural Network
3. Support Vector Machine
4. K Nearest Neighbor

- **Unsupervised learning**

1. K-means Clustering
2. Hierarchical Clustering

# Model representation

- Example: We want to find out the relationships between housing prices and size in US. The known data are drawn using x.
- The model representing these data is shown using the blue line.



# Model representation

Training set of housing prices

**m** = Number of training examples

(The number of **x**)

- $(x, y)$ : one training example
- $(x^{(i)}, y^{(i)})$ : the  $i$ -th training example

**x** = “input” variable

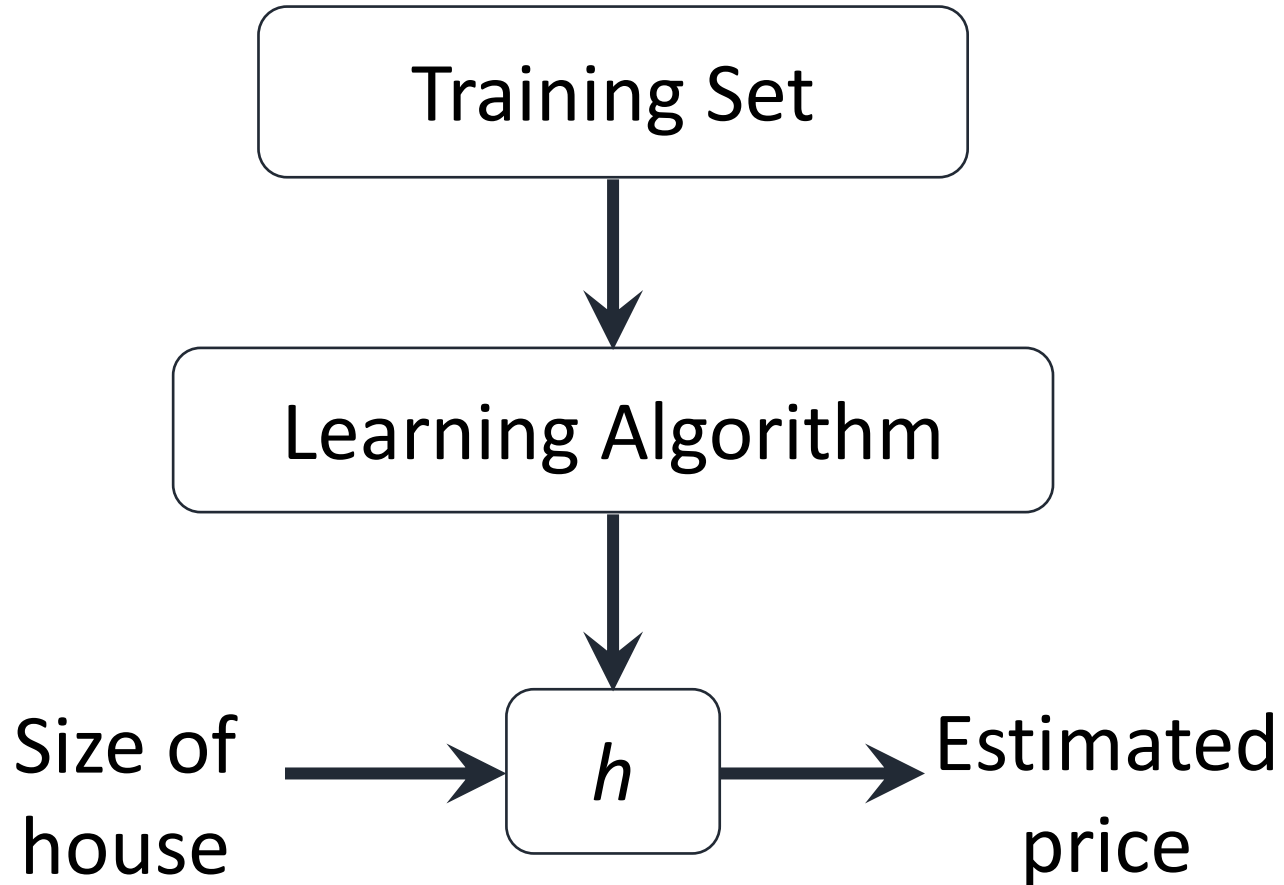
- $x^{(1)}=2104, x^{(2)}=1416, \dots$

**y** = “output” variable

- $y^{(1)}=460, y^{(2)}=232, \dots$

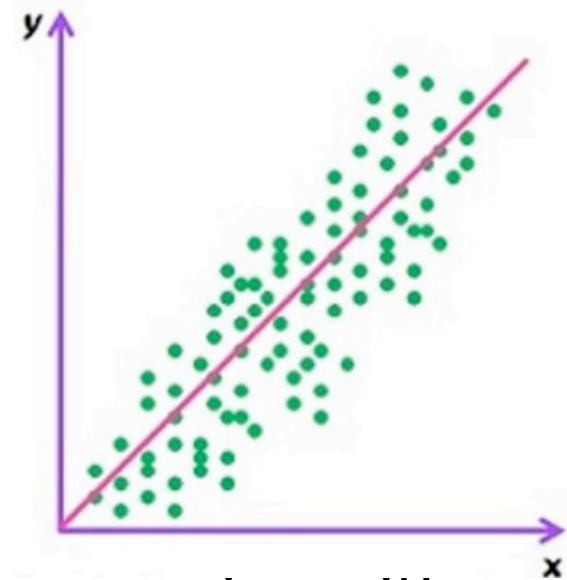
Size in feet <sup>2</sup> (x)	Price (\$) in 1000's (y)
2104	460
1416	232
1534	315
852	178
...	...

# Model representation



**How do we represent  $h$  ?**

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$



Linear regression with one variable.  
Univariate linear regression.

# Cost Function

Hypothesis:

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

How to calculate  
 $\theta_0$  and  $\theta_1$ ?

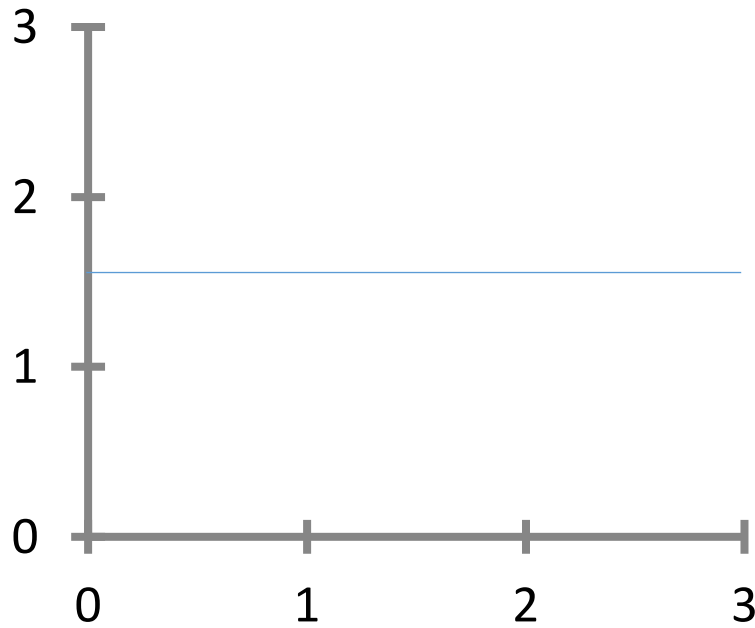
Training Set (m=42)

Size in feet <sup>2</sup> (x)	Price (\$) in 1000's (y)
2104	460
1416	232
1534	315
852	178
...	...

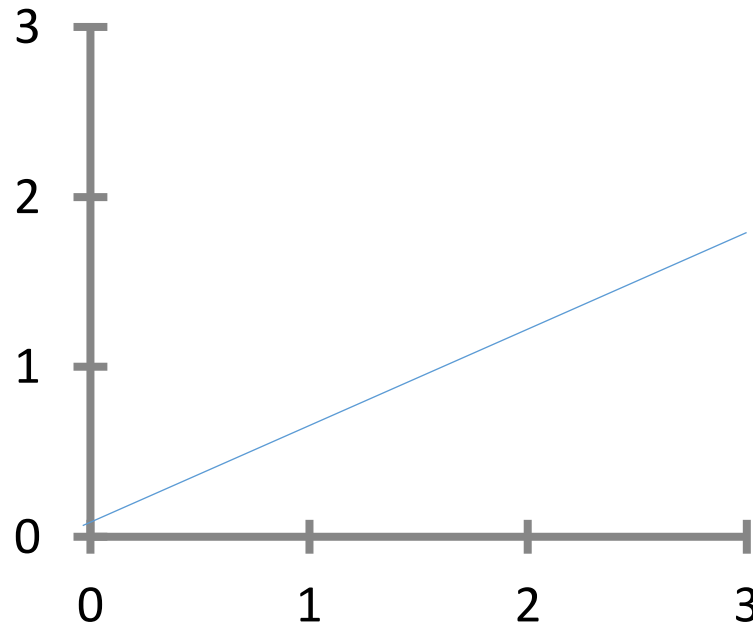
# Cost Function

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

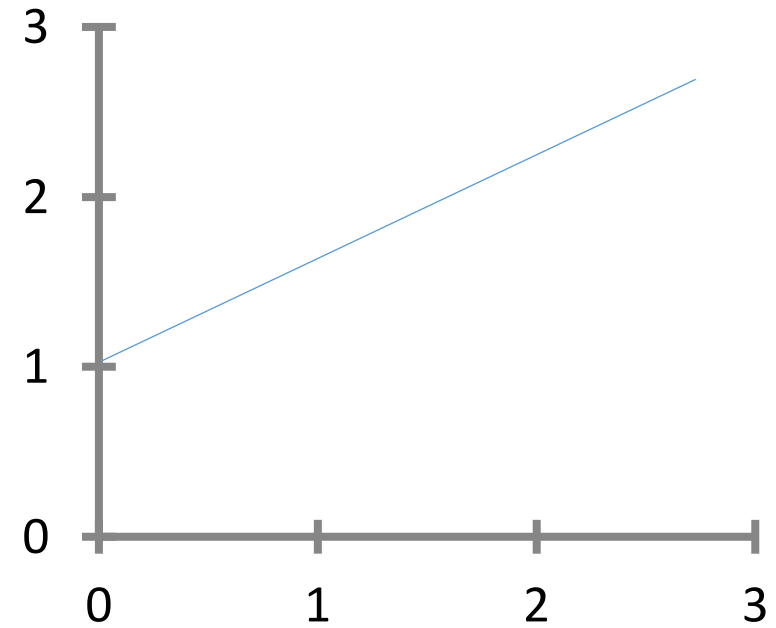
Idea: Choose  $\theta_0, \theta_1$  so that  $h_{\theta}(x)$  is close to  $y$  for our training examples



$$\theta_0 = 1.5$$
$$\theta_1 = 0$$

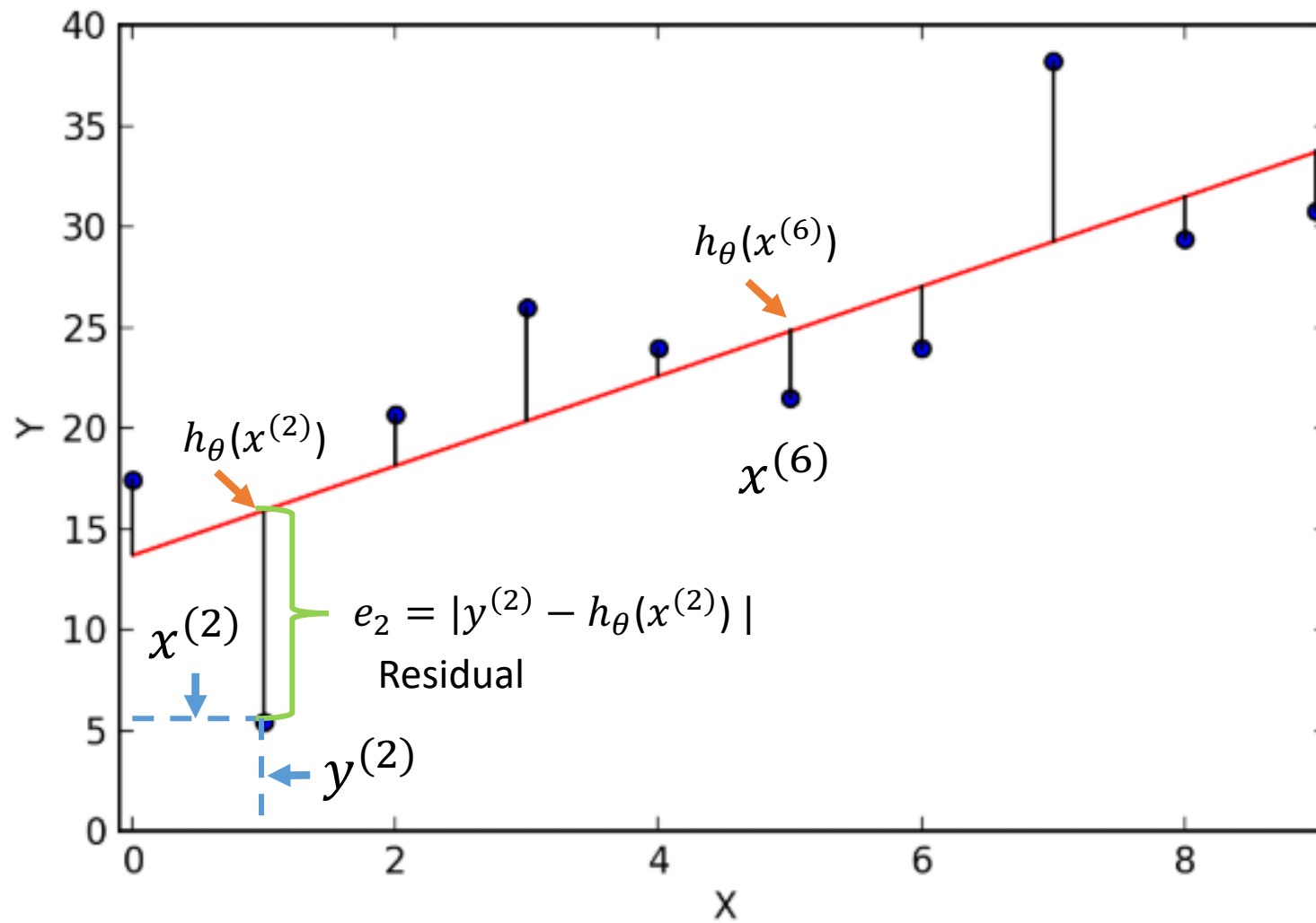


$$\theta_0 = 0$$
$$\theta_1 = 0.5$$



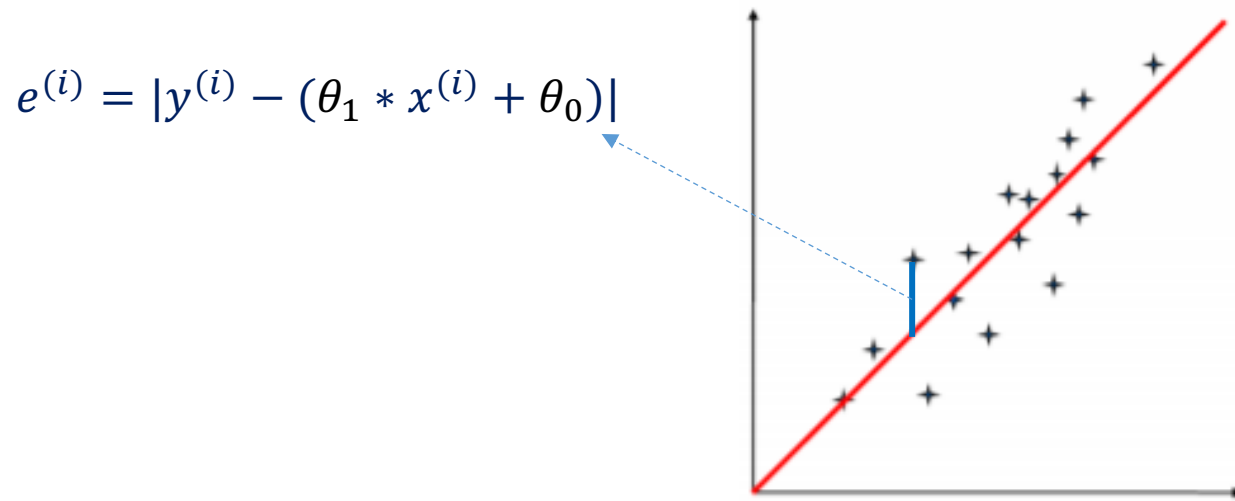
$$\theta_0 = 1$$
$$\theta_1 = 0.5$$

# Residual





# Cost function



- Error for a data sample  $(x^{(i)}, y^{(i)})$  is  $e^{(i)} = |y_i - (\theta_1 * x^{(i)} + \theta_0)|$
- The **sum of squared errors** is:

$$SSE = \sum_{i=1}^m (y^{(i)} - (\theta_1 * x^{(i)} + \theta_0))^2 = \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

- The objective is to find the best fitting (i.e., minimizing the sum of squared errors)

Cost function

Hypothesis:  $h_{\theta}(x) = \theta_0 + \theta_1 x$

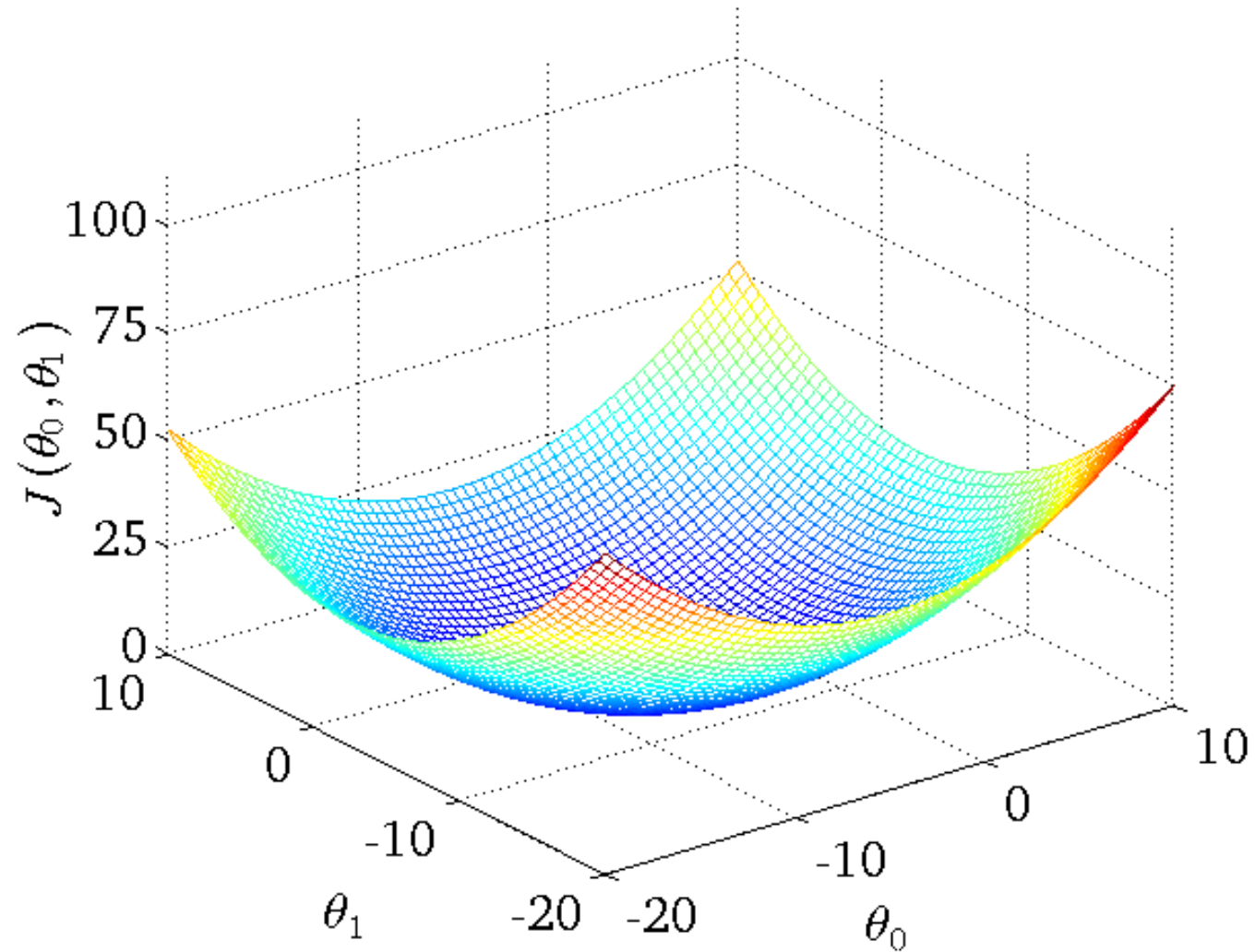
Parameters:  $\theta_0, \theta_1$

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

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

# Cost function

$$J(\theta_0, \theta_1)$$



# Gradient descent

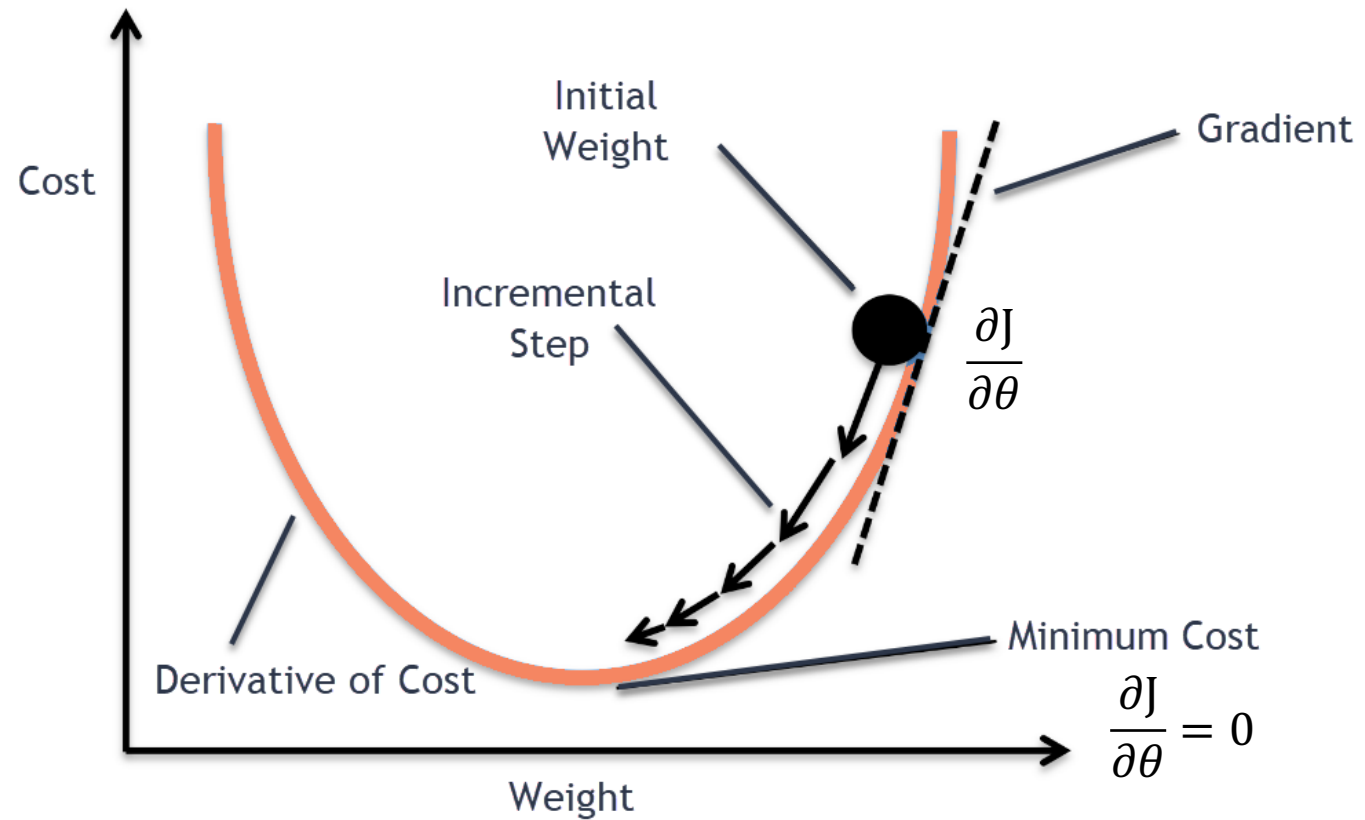
Have some function  $J(\theta_0, \theta_1)$

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

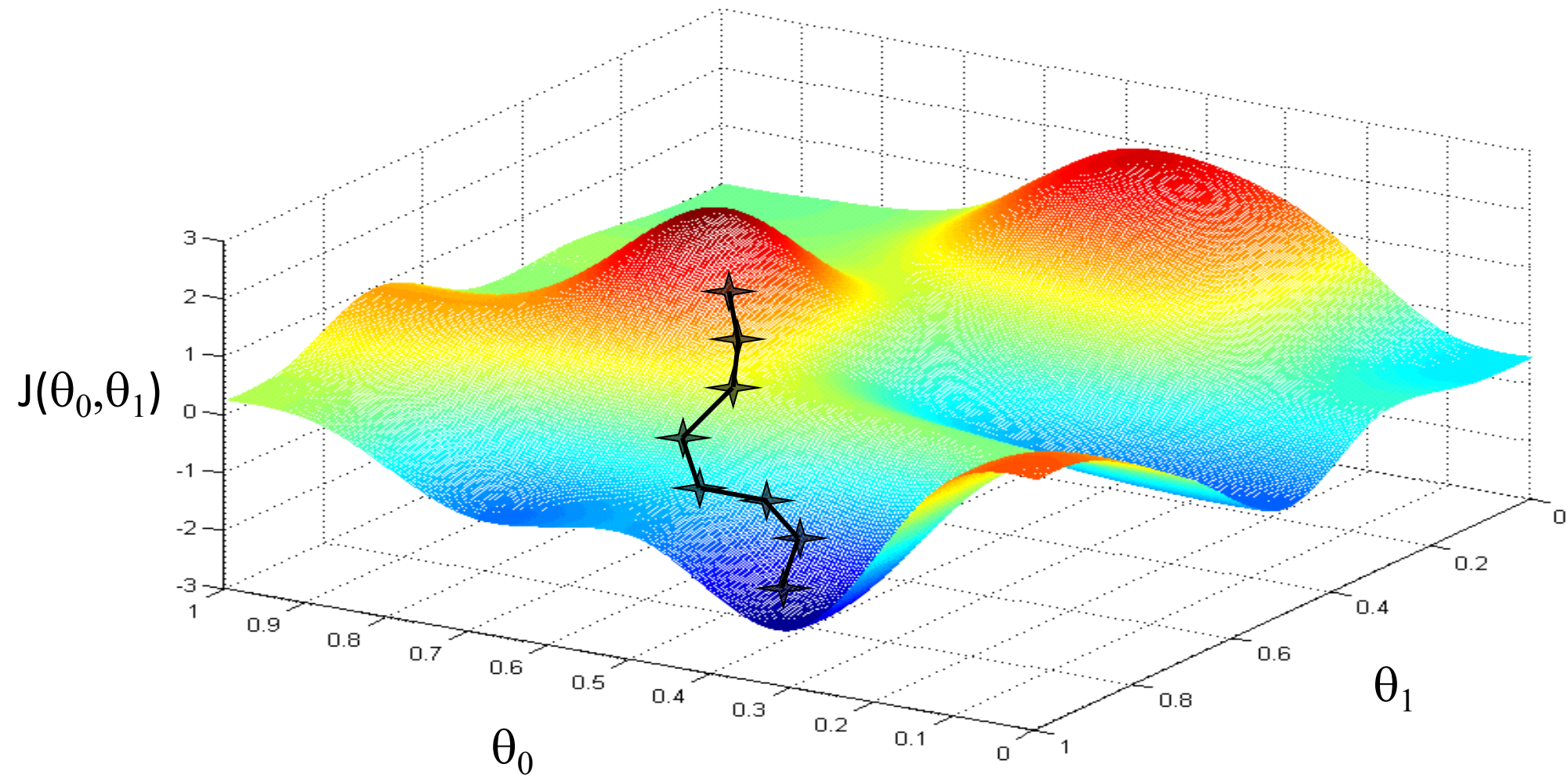
## Outline:

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

# Gradient descent



# Gradient descent



# Gradient descent

repeat until convergence {

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

}

---

Correct: **Simultaneous** update

$$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$

$$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$

$$\theta_0 := \text{temp0}$$

$$\theta_1 := \text{temp1}$$

Incorrect:

$$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$

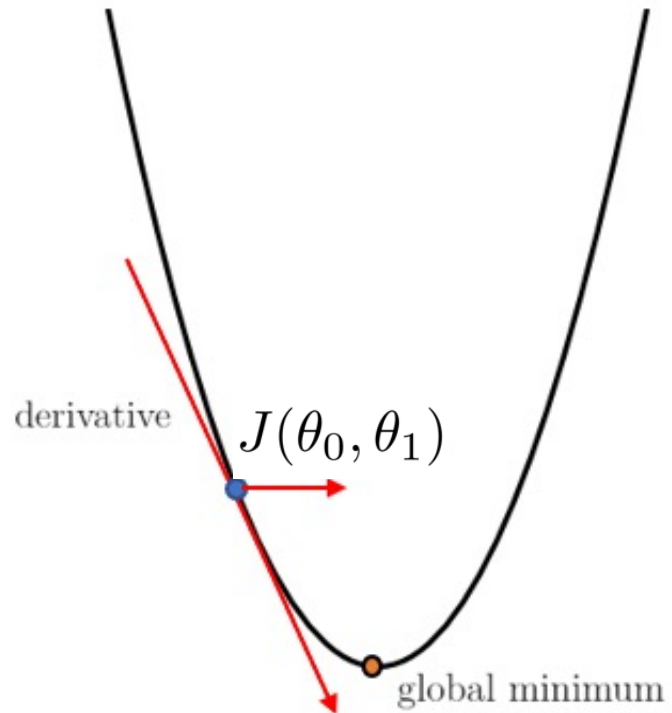
$$\theta_0 := \text{temp0}$$

$$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$

$$\theta_1 := \text{temp1}$$

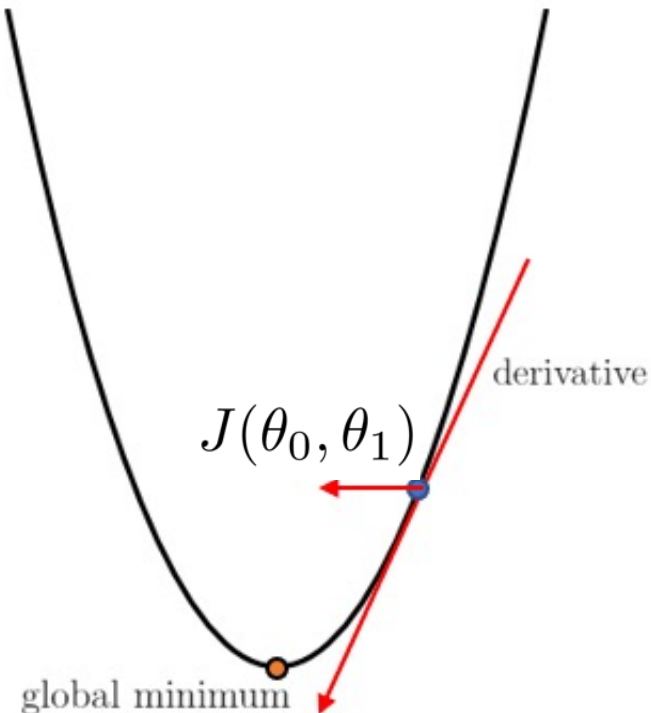
# Gradient descent

Before minimum



Since the derivative is negative,  
if we subtract the derivative from  $\theta$   
it will increase and go closer the minimum.

After minimum

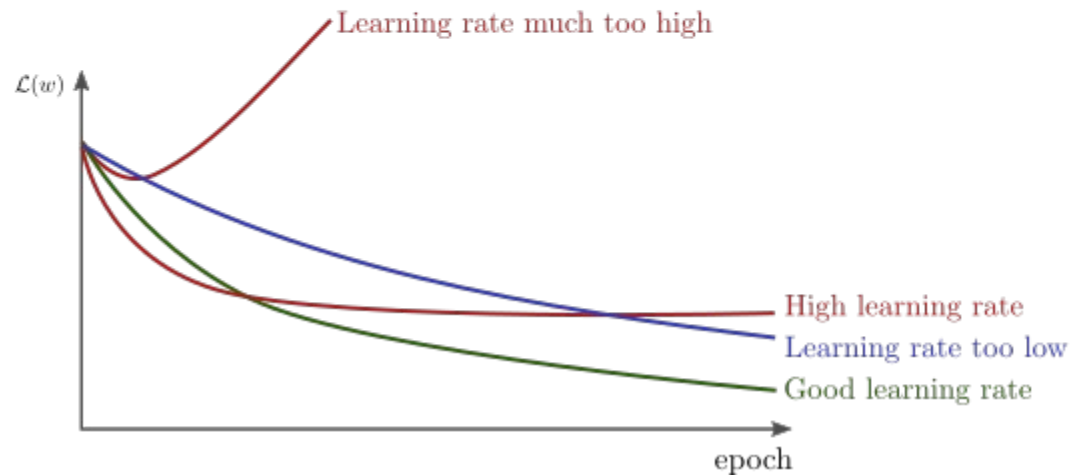
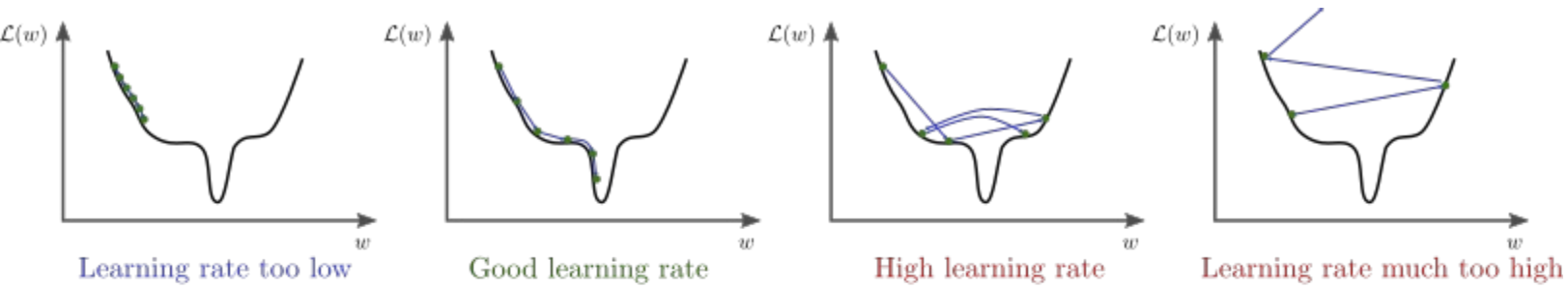


Since the derivative is positive,  
if we subtract the derivative from  $\theta$   
it will decrease and go closer the minimum.



# Learning rate

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



- If  $\alpha$  is too small, gradient descent can be slow.
- If  $\alpha$  is too large, gradient descent can overshoot the minimum. It may fail to converge, or even diverge.

# Gradient descent on linear regression

## Gradient descent algorithm

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

## Linear Regression Model

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

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

# Gradient descent on linear regression

repeat until convergence {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) \cdot x^{(i)}$$

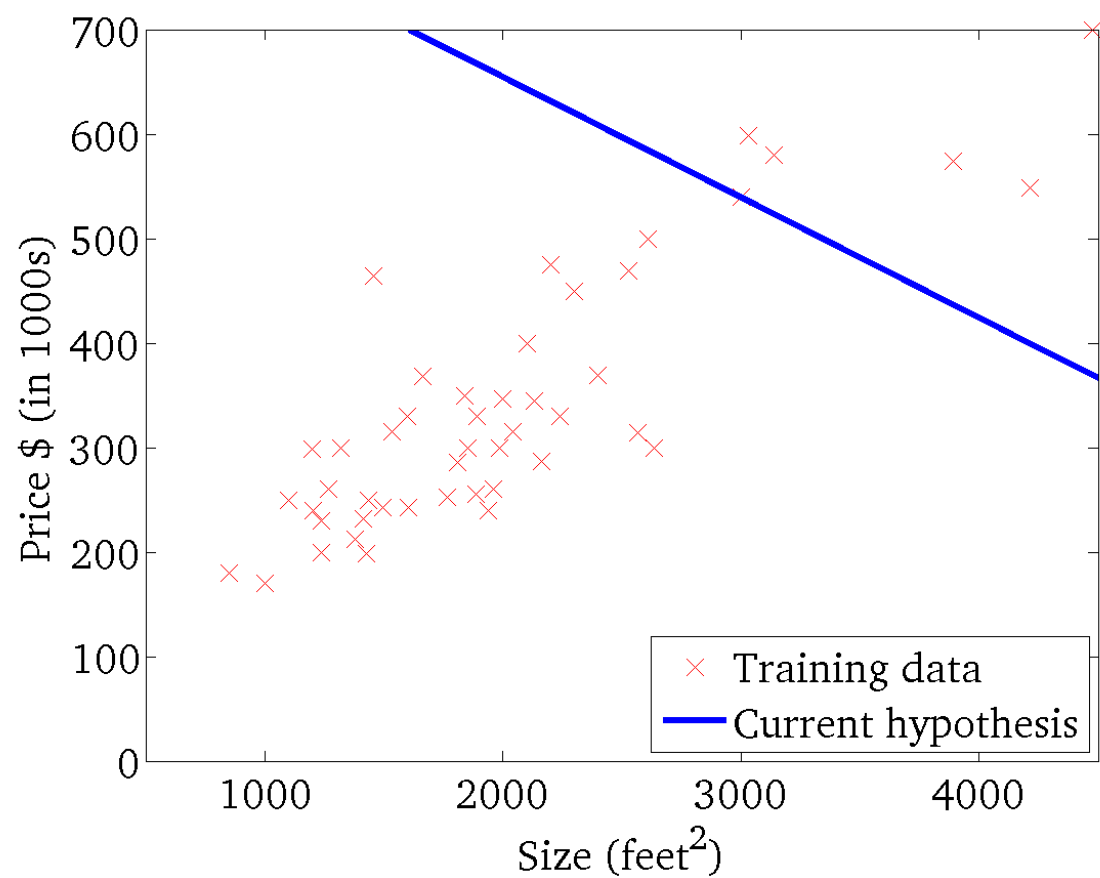
} update  
 $\theta_0$  and  $\theta_1$   
simultaneously

}

# Gradient descent on linear regression

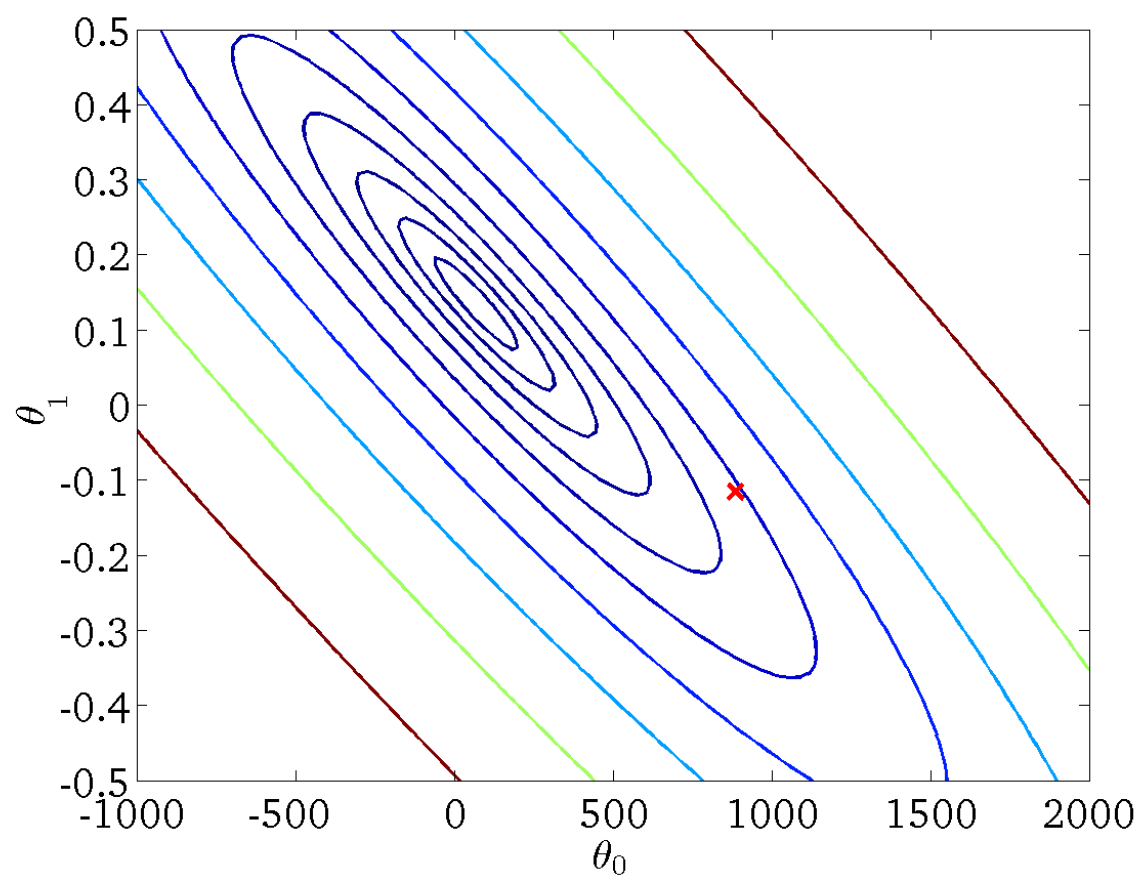
$$h_{\theta}(x)$$

(for fixed  $\theta_0, \theta_1$ , this is a function of  $x$ )



$$J(\theta_0, \theta_1)$$

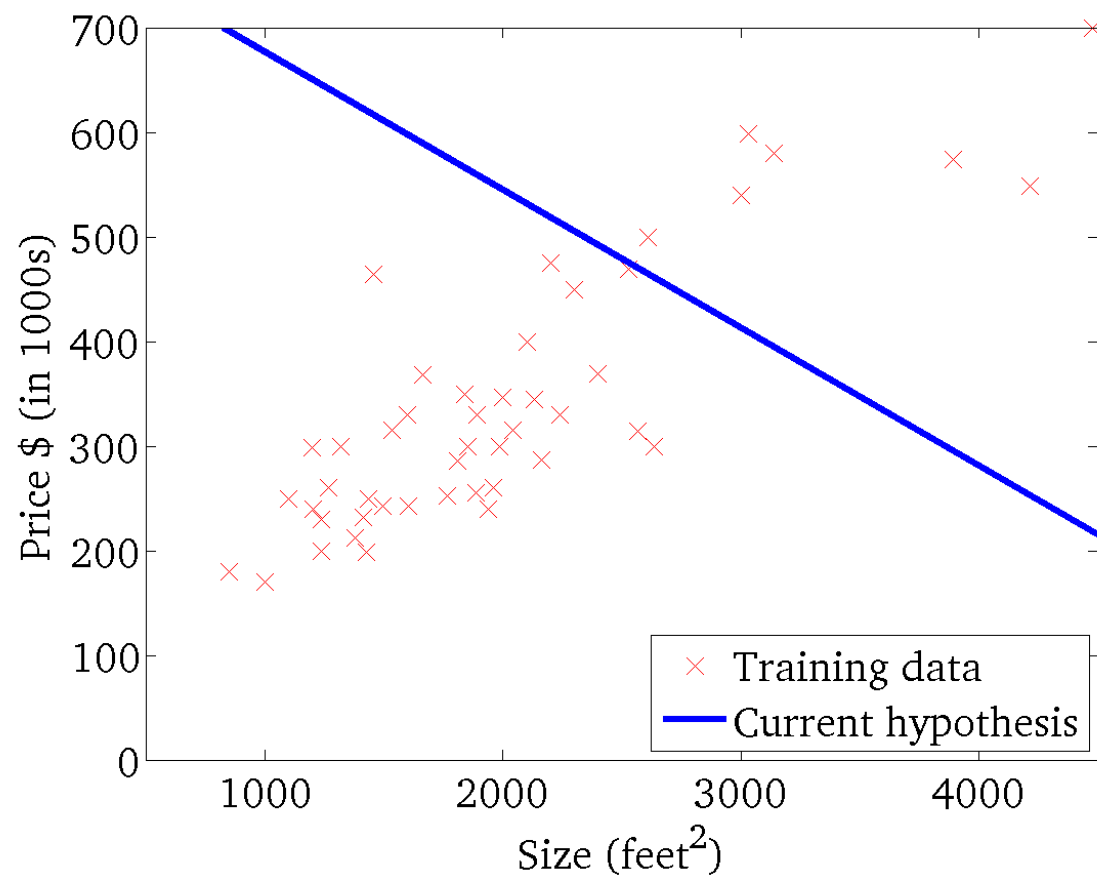
(function of the parameters  $\theta_0, \theta_1$ )



# Gradient descent on linear regression

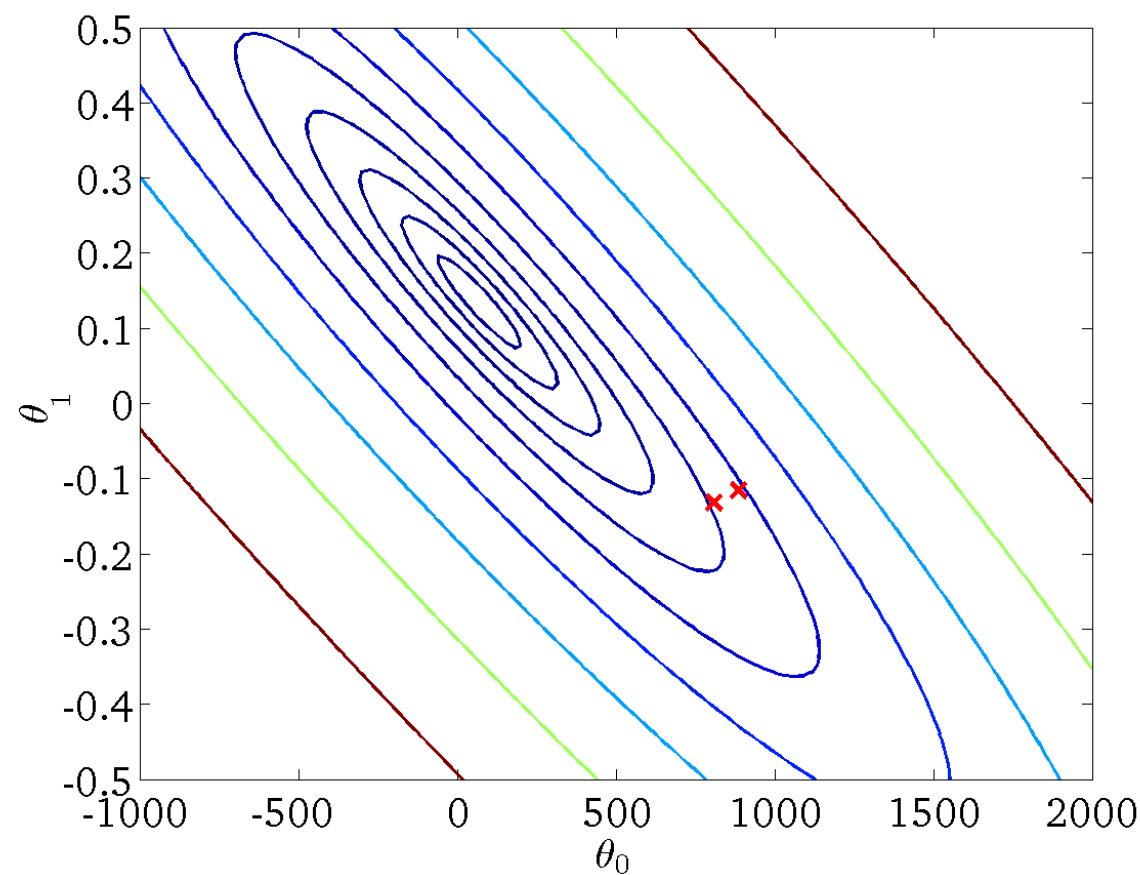
$$h_{\theta}(x)$$

(for fixed  $\theta_0, \theta_1$ , this is a function of  $x$ )



$$J(\theta_0, \theta_1)$$

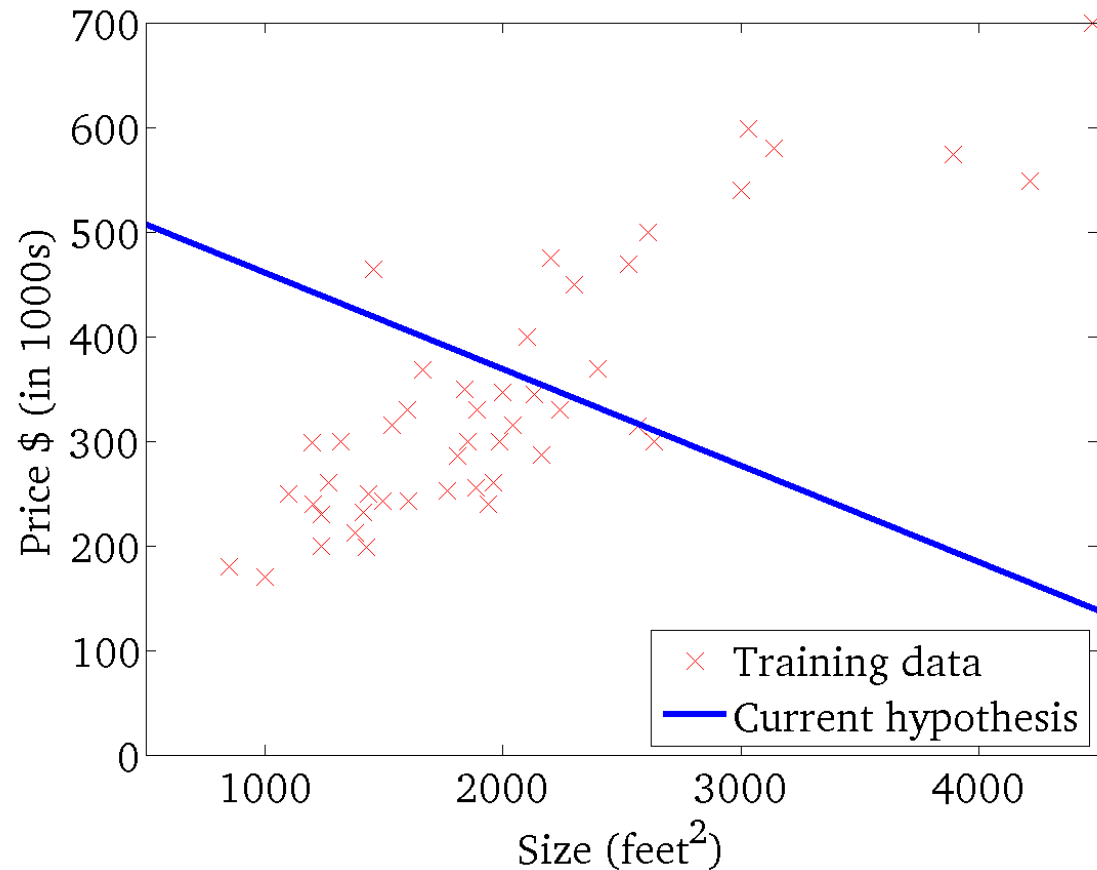
(function of the parameters  $\theta_0, \theta_1$ )



# Gradient descent on linear regression

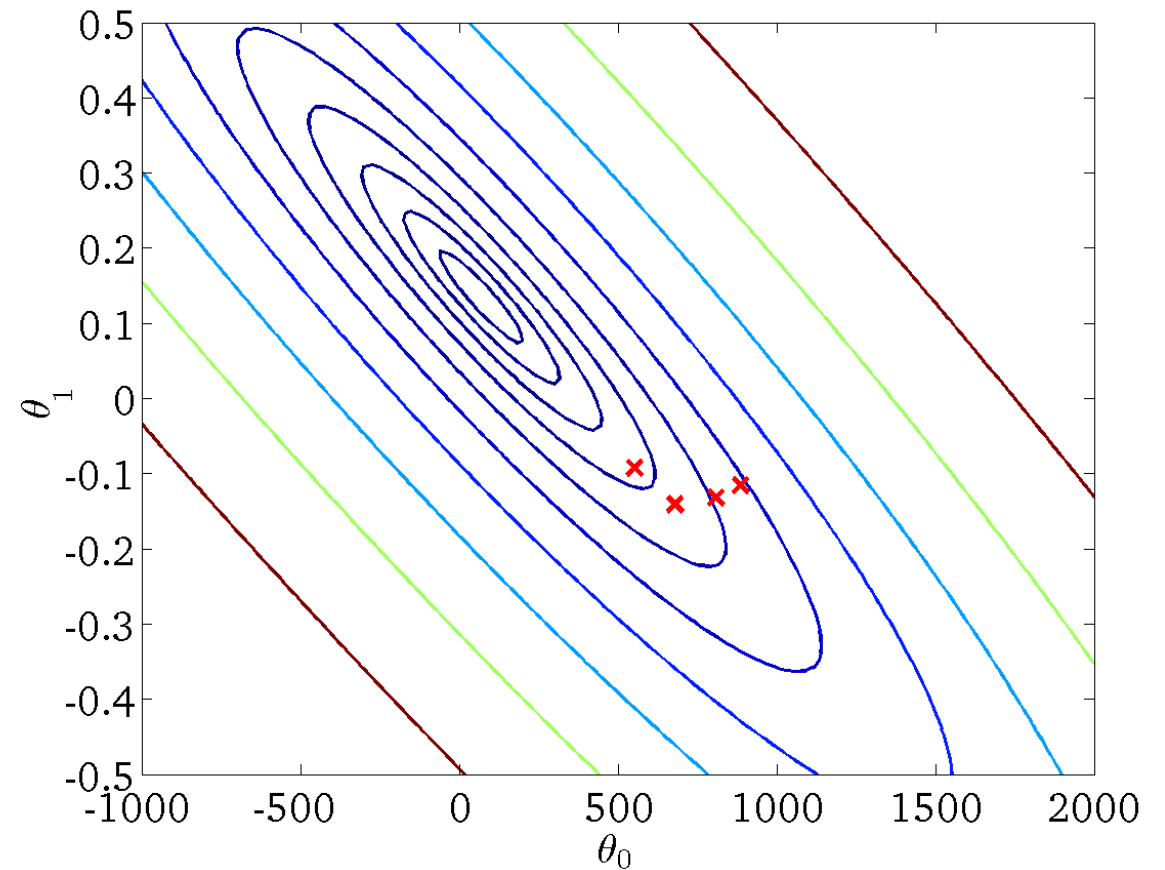
$$h_{\theta}(x)$$

(for fixed  $\theta_0, \theta_1$ , this is a function of  $x$ )



$$J(\theta_0, \theta_1)$$

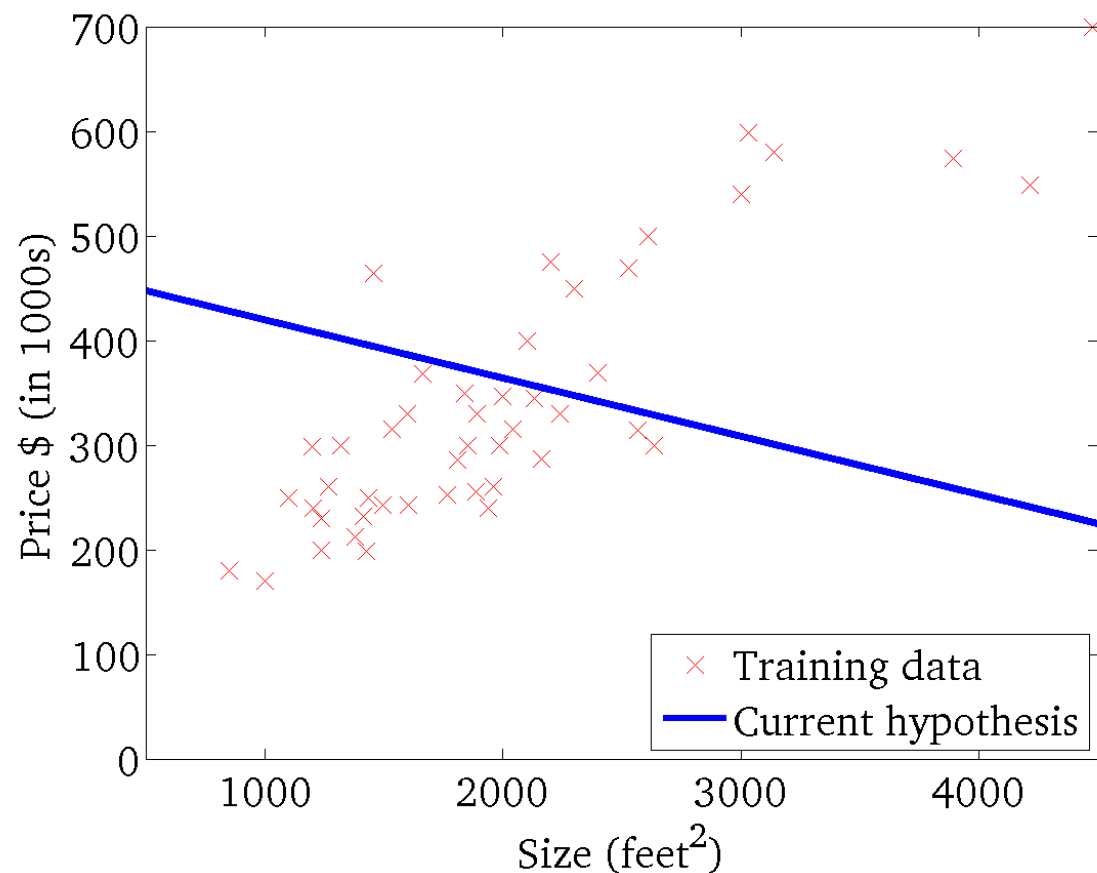
(function of the parameters  $\theta_0, \theta_1$ )



# Gradient descent on linear regression

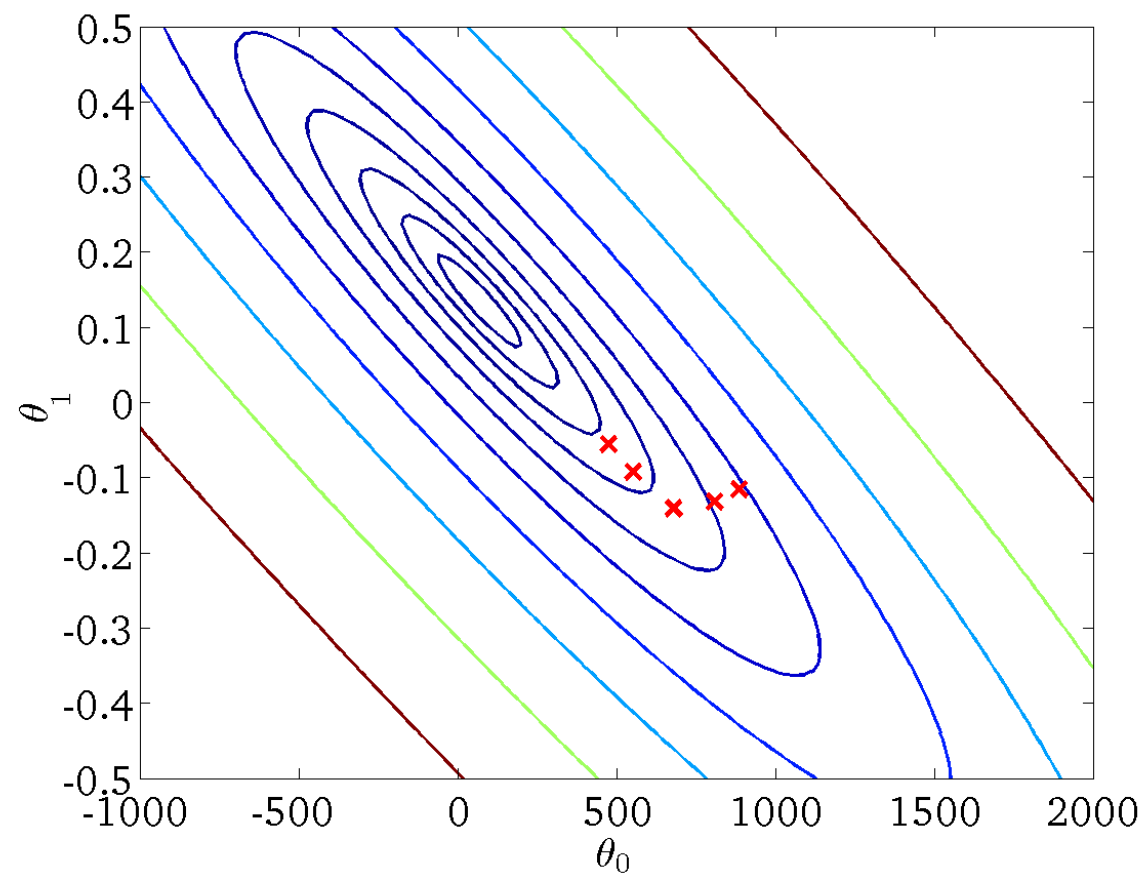
$$h_{\theta}(x)$$

(for fixed  $\theta_0, \theta_1$ , this is a function of  $x$ )



$$J(\theta_0, \theta_1)$$

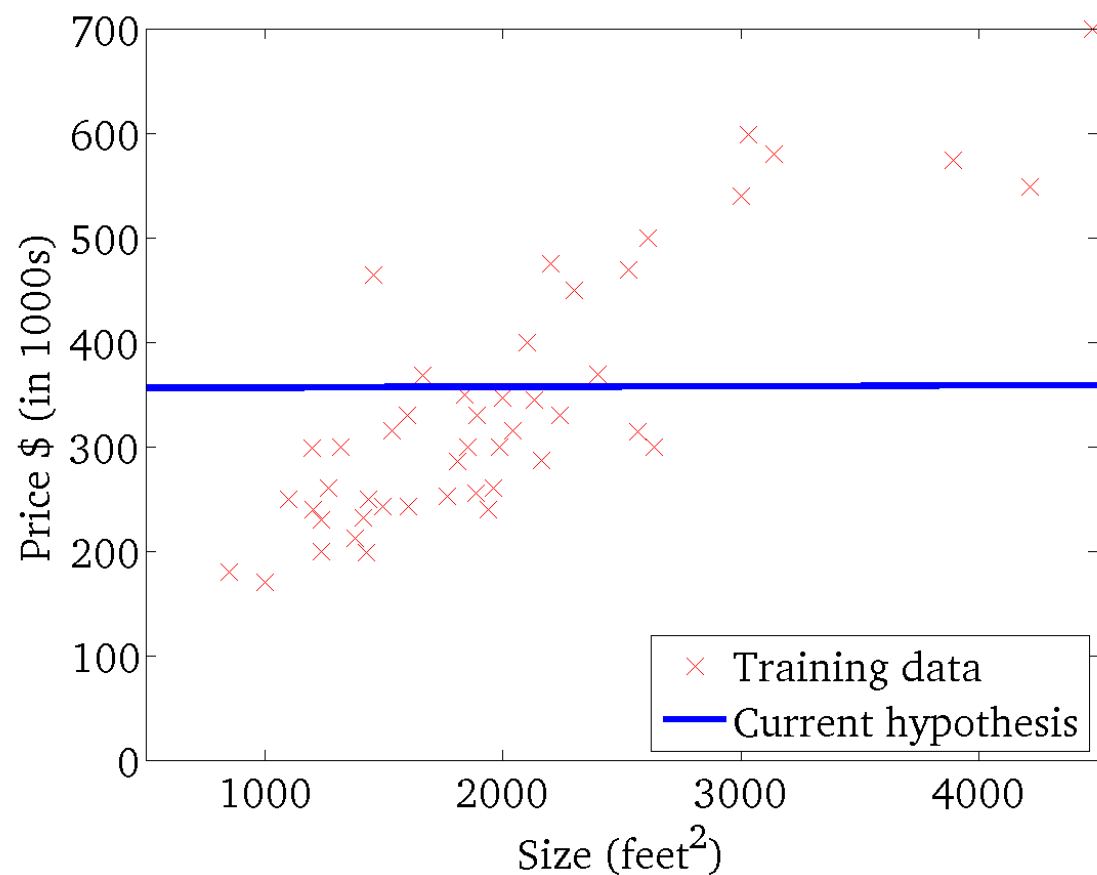
(function of the parameters  $\theta_0, \theta_1$ )



# Gradient descent on linear regression

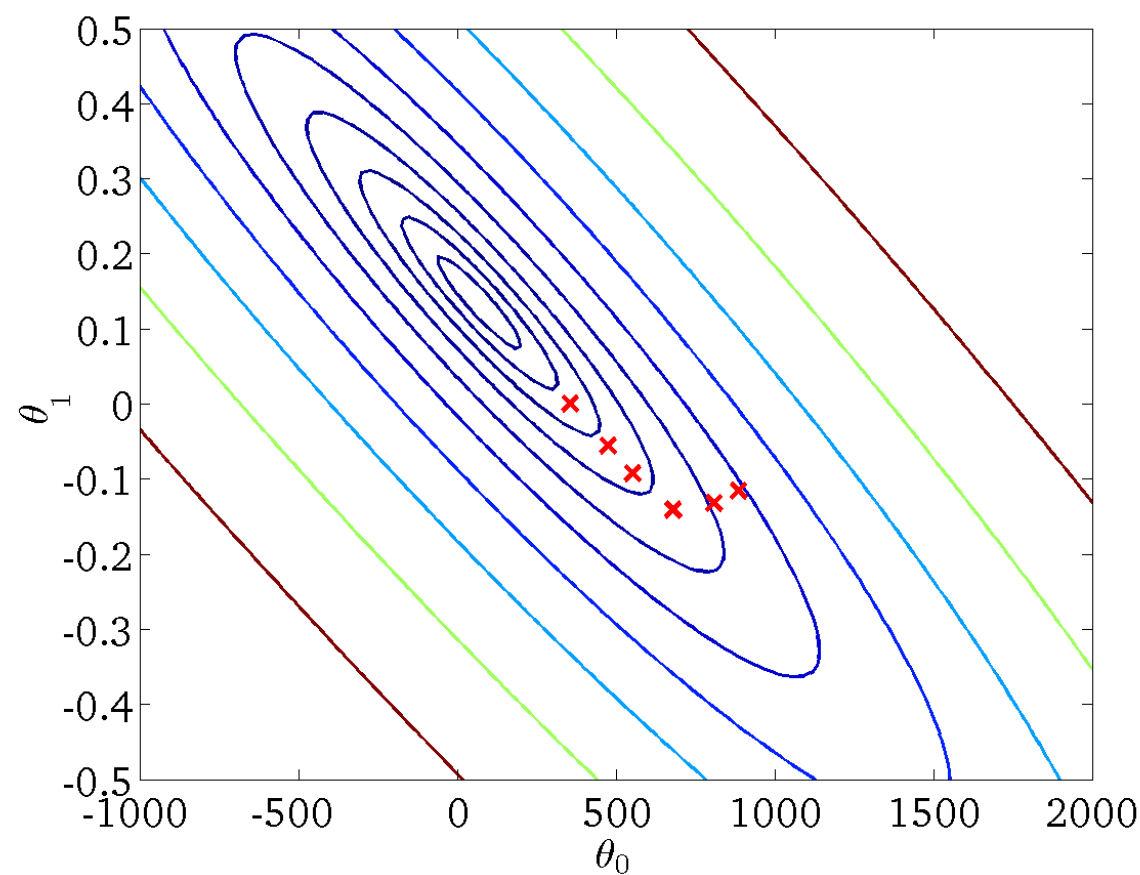
$$h_{\theta}(x)$$

(for fixed  $\theta_0, \theta_1$ , this is a function of  $x$ )



$$J(\theta_0, \theta_1)$$

(function of the parameters  $\theta_0, \theta_1$ )

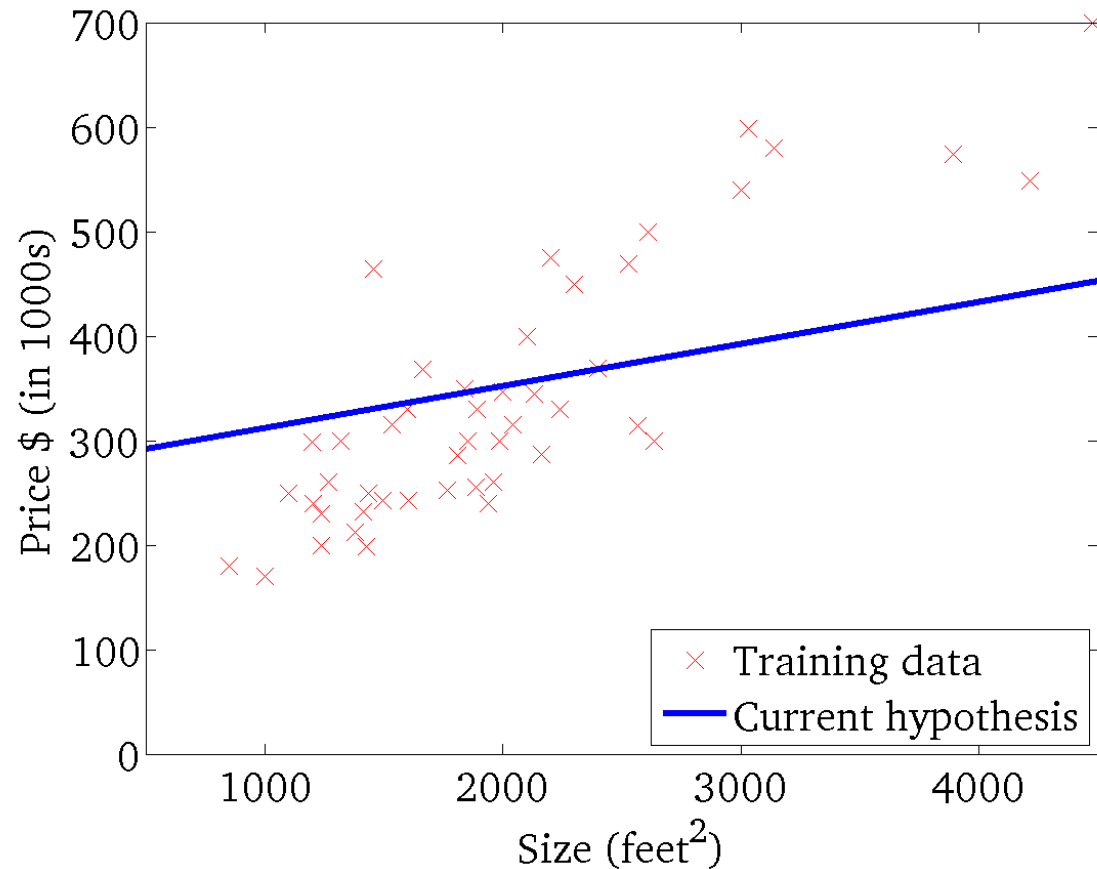




# Gradient descent on linear regression

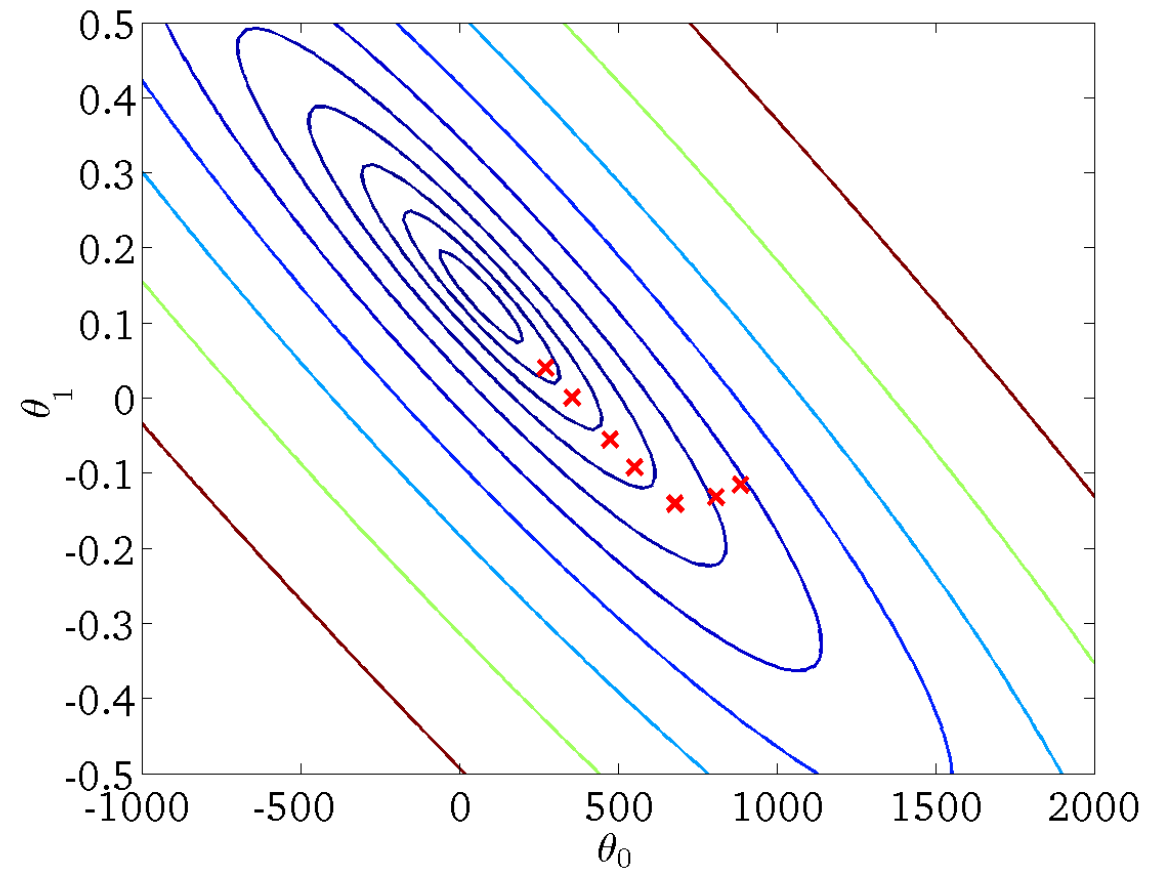
$$h_{\theta}(x)$$

(for fixed  $\theta_0, \theta_1$ , this is a function of  $x$ )



$$J(\theta_0, \theta_1)$$

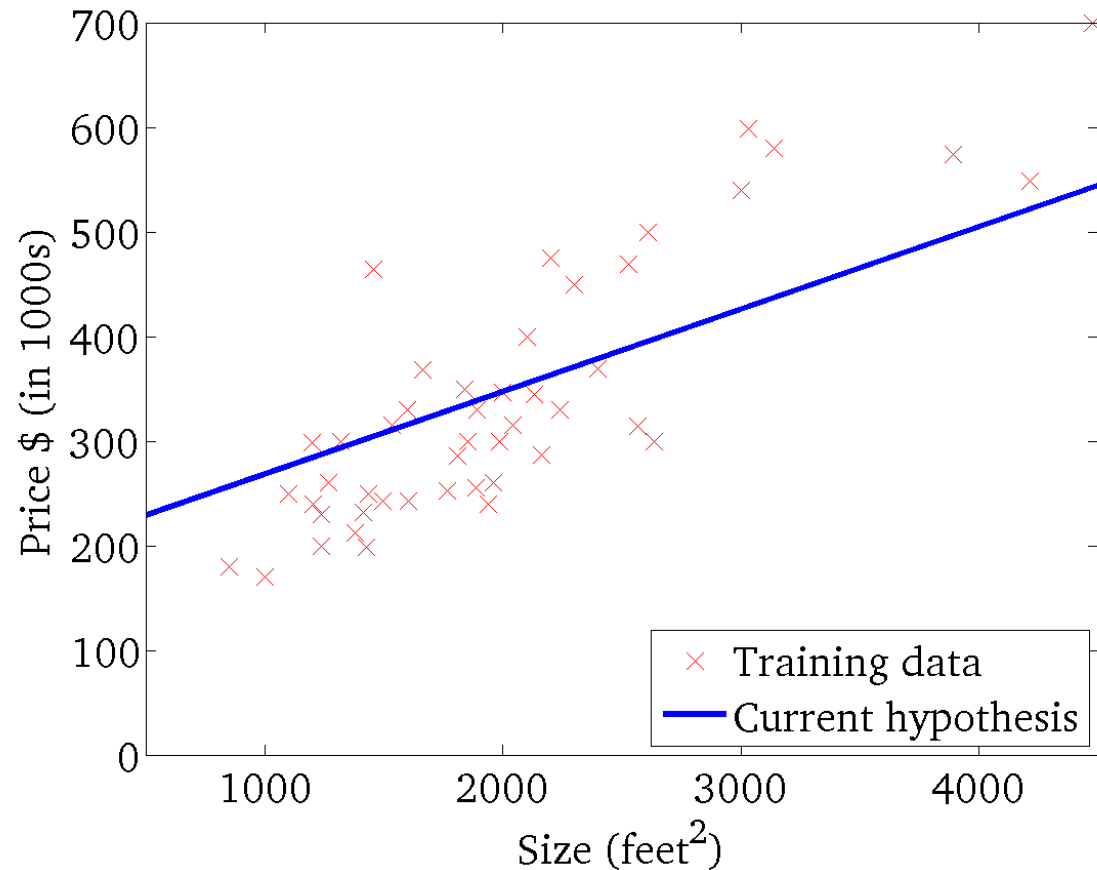
(function of the parameters  $\theta_0, \theta_1$ )



# Gradient descent on linear regression

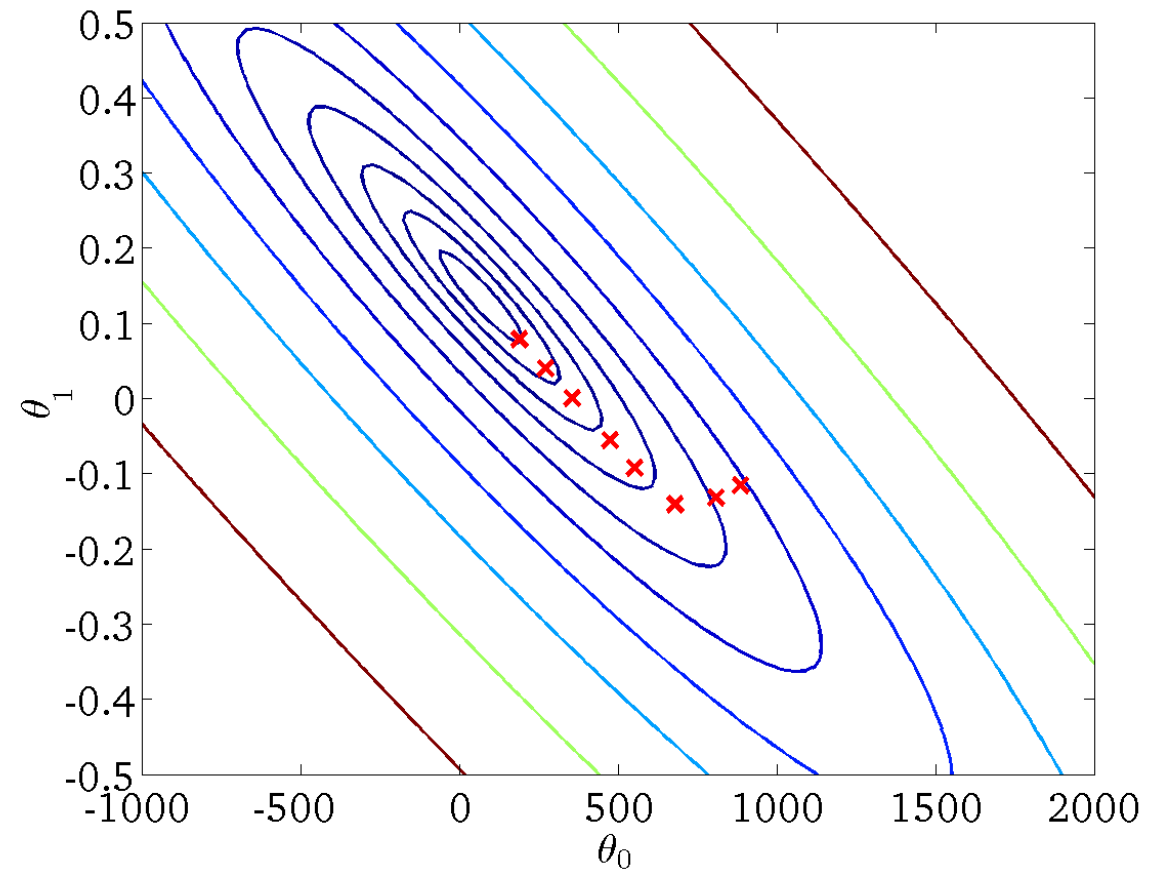
$$h_{\theta}(x)$$

(for fixed  $\theta_0, \theta_1$ , this is a function of  $x$ )



$$J(\theta_0, \theta_1)$$

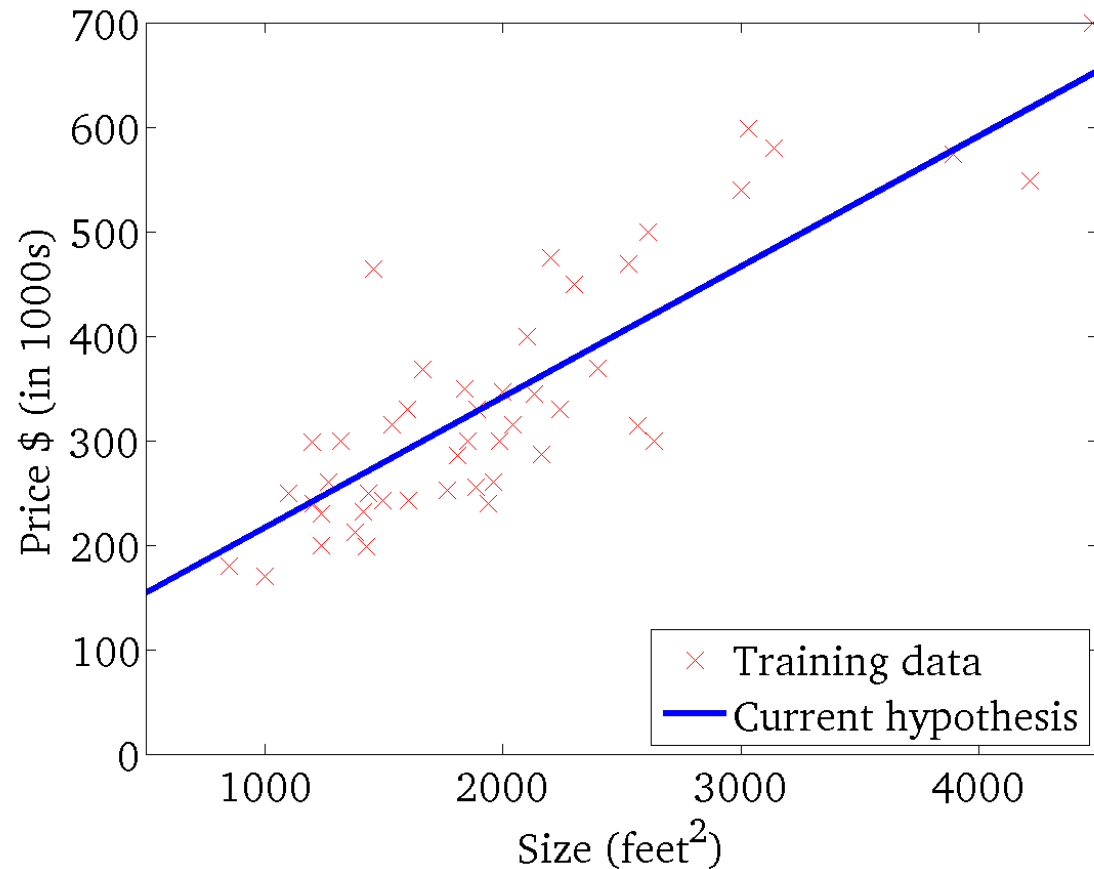
(function of the parameters  $\theta_0, \theta_1$ )



# Gradient descent on linear regression

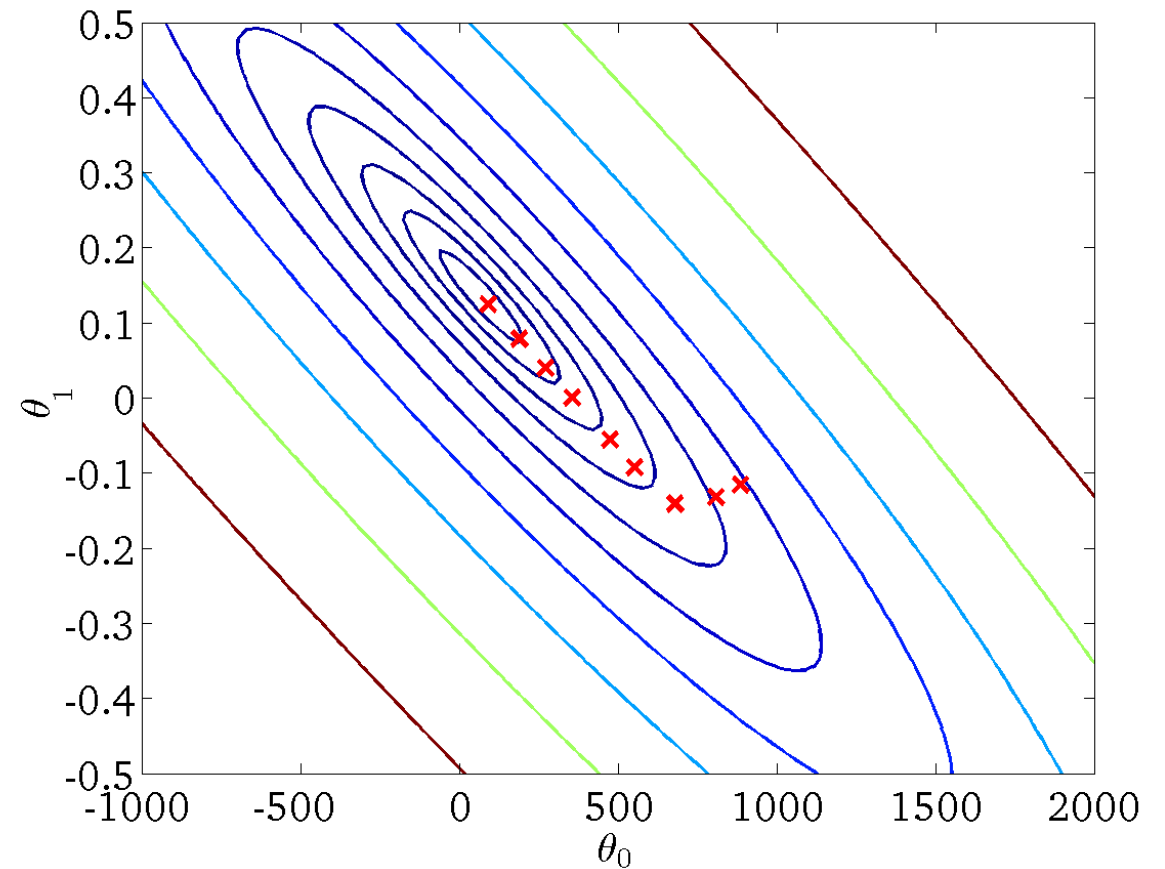
$$h_{\theta}(x)$$

(for fixed  $\theta_0, \theta_1$ , this is a function of  $x$ )



$$J(\theta_0, \theta_1)$$

(function of the parameters  $\theta_0, \theta_1$ )



# Derive the Solution for Linear Regression on One-Dimensional Data

$$\theta_1 = \frac{\sum (x^{(i)} - \bar{x})(y^{(i)} - \bar{y})}{\sum (x^{(i)} - \bar{x})^2}$$

$$\theta_0 = \bar{y} - \theta_1 * \bar{x}$$

# Derive the Solution for Linear Regression on One-Dimensional Data (optional)

- $\theta_1, \theta_0$  are estimated by minimizing the cost function.

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

- The optimal  $\theta_1, \theta_0$  must satisfy

$$\frac{\partial J}{\partial \theta_1} = 0 \Rightarrow \frac{1}{m} \sum_{i=1}^m ((\theta_1 x^{(i)} + \theta_0) - y^{(i)}) x^{(i)} = 0 \Rightarrow \sum_{i=1}^m y^{(i)} x^{(i)} - \theta_1 \sum_{i=1}^m (x^{(i)})^2 - \theta_0 \sum_{i=1}^m x^{(i)} = 0$$

$$\frac{\partial J}{\partial \theta_0} = 0 \Rightarrow \frac{1}{m} \sum_{i=1}^m ((\theta_1 x^{(i)} + \theta_0) - y^{(i)}) = 0 \Rightarrow \sum_{i=1}^m y^{(i)} - \theta_1 \sum_{i=1}^m x^{(i)} - \sum_{i=1}^m \theta_0 = 0 \Rightarrow \theta_0 = \bar{y} - \theta_1 \bar{x}$$

$$\sum_{i=1}^m y^{(i)} x^{(i)} - \theta_1 \sum_{i=1}^m (x^{(i)})^2 - (\bar{y} - \theta_1 \bar{x}) \sum_{i=1}^m x^{(i)} = 0 \Rightarrow \theta_1 = \frac{\bar{y} \sum_{i=1}^m x^{(i)} - \sum_{i=1}^m y^{(i)} x^{(i)}}{\bar{x} \sum_{i=1}^m x^{(i)} - \sum_{i=1}^m (x^{(i)})^2} = \frac{\sum_{i=1}^m (x^{(i)} - \bar{x})(y^{(i)} - \bar{y})}{\sum_{i=1}^m (x^{(i)} - \bar{x})^2}$$

# Example

Student	Test score	IQ	Study hours
1	100	110	40
2	90	120	30
3	80	100	20
4	70	90	0
5	60	80	10

develop a regression equation to predict test scores (y), based on students' IQs (x).

$$y = \theta_0 + \theta_1 x$$

# Solution

$$\theta_1 = \frac{\sum (x^{(i)} - \bar{x})(y^{(i)} - \bar{y})}{\sum (x^{(i)} - \bar{x})^2}$$

$$\bar{x} = 100, \bar{y} = 80$$

$$\begin{aligned}\theta_1 &= \frac{(110 - 100)(100 - 80) + (120 - 100)(90 - 80) + (100 - 100)(80 - 80) + (70 - 80)(90 - 100) + (60 - 80)(80 - 100)}{(110 - 100)^2 + (120 - 100)^2 + (100 - 100)^2 + (90 - 100)^2 + (80 - 100)^2} \\ &= \frac{200 + 200 + 100 + 400}{100 + 400 + 100 + 400} = \frac{900}{1000} = 0.9\end{aligned}$$

$$\theta_0 = 80 - 0.9 * 100 = -10$$

$$y = -10 + 0.9x$$

Test Score vs IQ plot

