

Linear Regression

Fitting a **line** through data points

Faculty of Mathematics and Computer Science, University of Bucharest
and
Sparktech Software

Academic Year 2018/2019, 1st Semester

History of Linear Regression

- ***"Nouvelles méthodes pour la détermination des orbites des comètes"***

Legendre, Adrien-Marie, 1805

- The first clear and concise exposition of the "least squares method"

- ***"Regression Towards Mediocrity in Hereditary Stature"***

Francis Galton, 1886

- Children with tall parents tend to be shorter... They "regressed" towards the mean

- ***"The Law of Ancestral Heredity"***

Karl Pearson, G. U. Yule, Norman Blanchard and Alice Lee, 1903

- ***"The goodness of fit of regression formulae, and the distribution of regression coefficients"***

Sir R.A. Fisher, 1922

Linear Regression Objectives

- Establish if there is a relationship between a **dependent variable** and one or more **independent variables**.

Linear Regression Objectives

- Establish if there is a relationship between a **dependent variable** and one or more **independent variables**.
 - The dependent variable is sometimes called the label, **target**, **response** or **output**
 - The independent variables are sometimes called features, **attributes** or **predictors**

Linear Regression Objectives

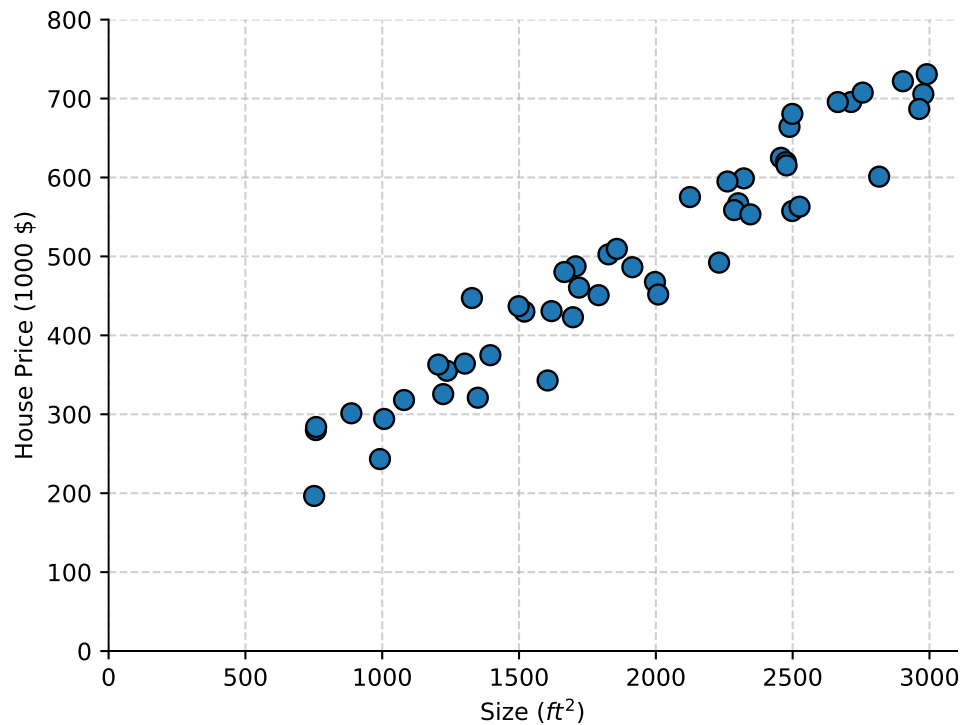
- Establish if there is a relationship between a **dependent variable** and one or more **independent variables**.
 - The dependent variable is sometimes called the label, **target**, **response** or **output**
 - The independent variables are sometimes called features, **attributes** or **predictors**
 - If there is only one independent variable, it is called *Simple Linear Regression*
 - If there are multiple independent variables, it is called *Multiple Linear Regression*

Linear Regression Objectives

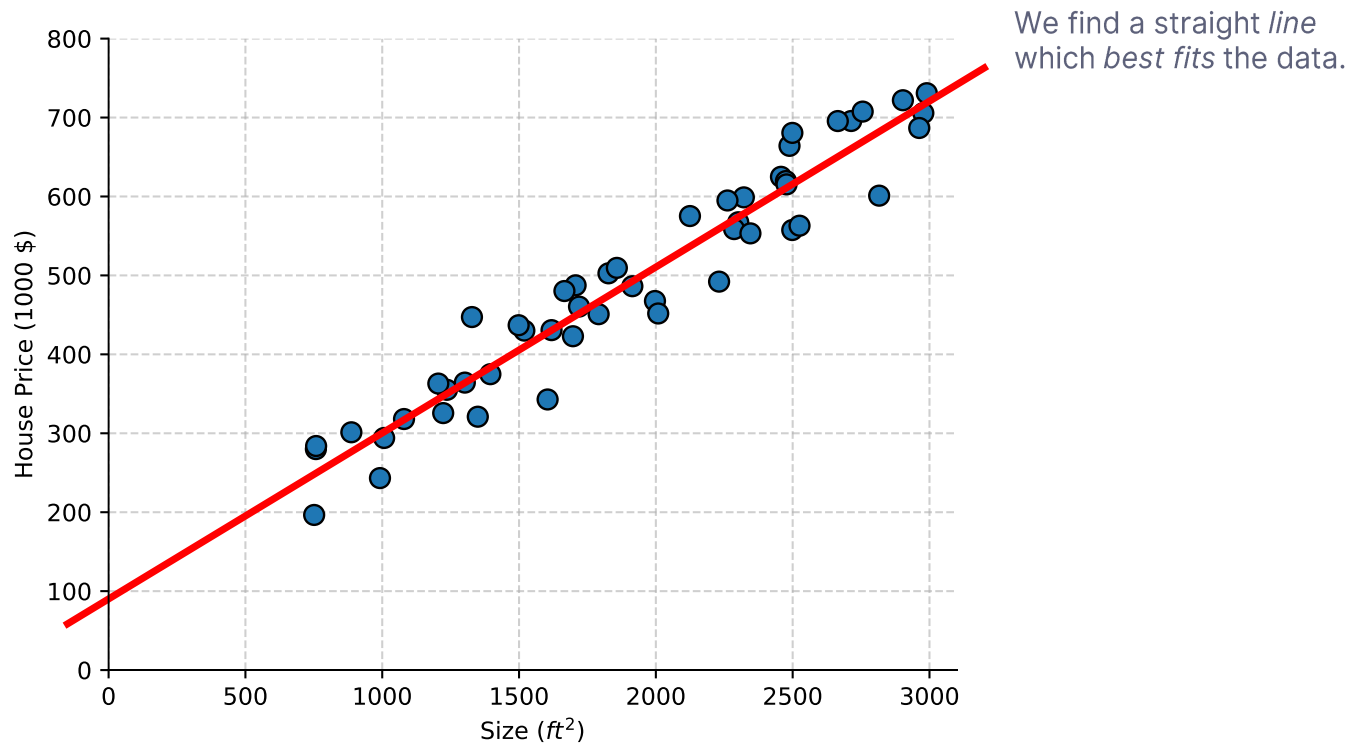
- Establish if there is a relationship between a **dependent variable** and one or more **independent variables**.
 - The dependent variable is sometimes called the label, **target**, **response** or **output**
 - The independent variables are sometimes called features, **attributes** or **predictors**
 - If there is only one independent variable, it is called *Simple Linear Regression*
 - If there are multiple independent variables, it is called *Multiple Linear Regression*
- **Forecast** new observations.
 - Use the established relationship to make *predictions* about new data.

Simple Linear Regression

Simple Linear Regression

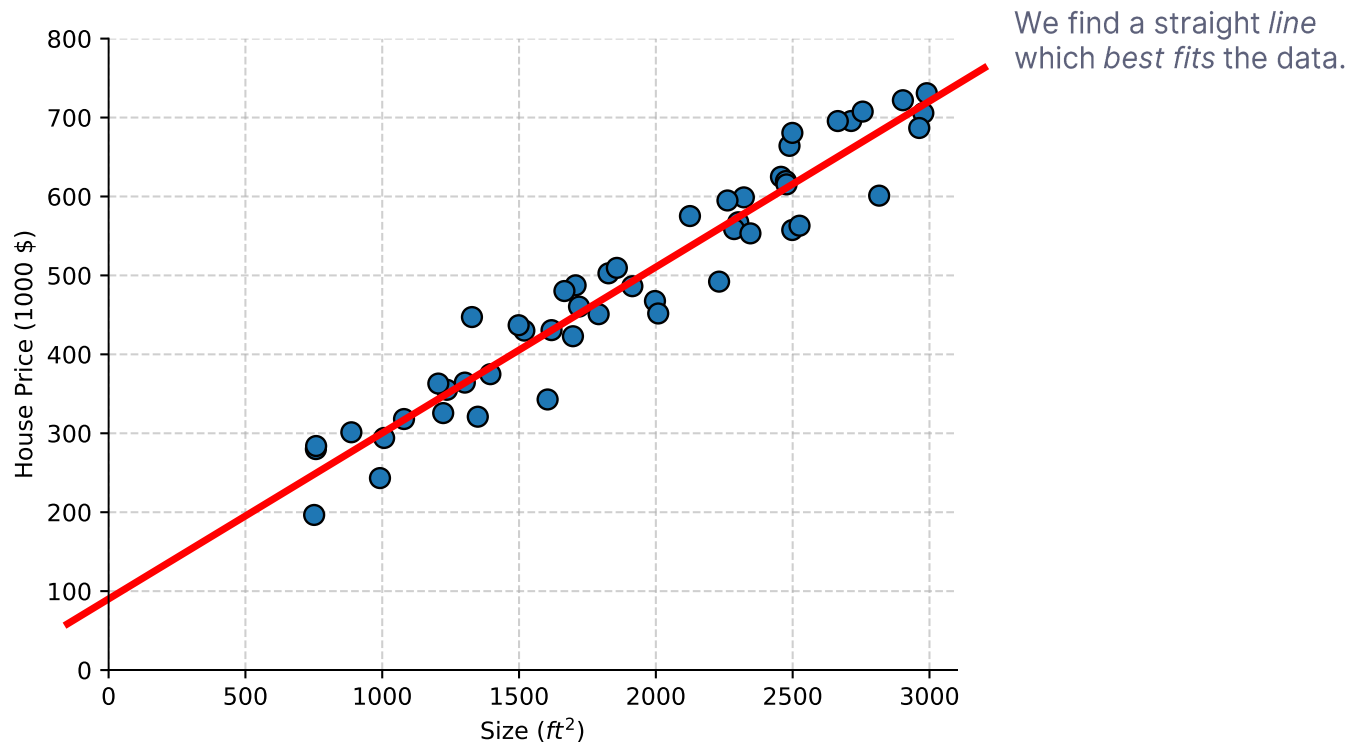


Simple Linear Regression



Simple Linear Regression

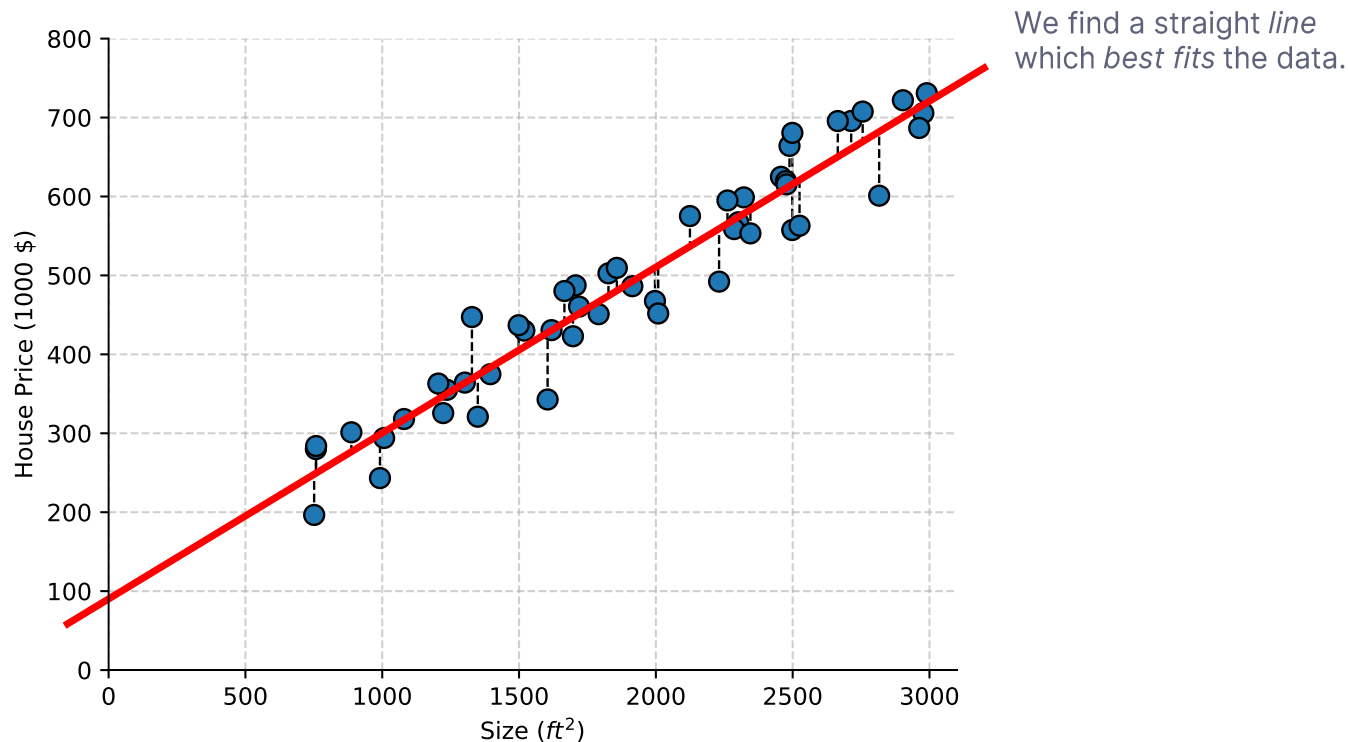
Slope is positive →
Positive correlation
between price and size
(i.e. price tends to grow
with size).



Simple Linear Regression

Slope is positive →
Positive correlation
between price and size
(i.e. price tends to grow
with size).

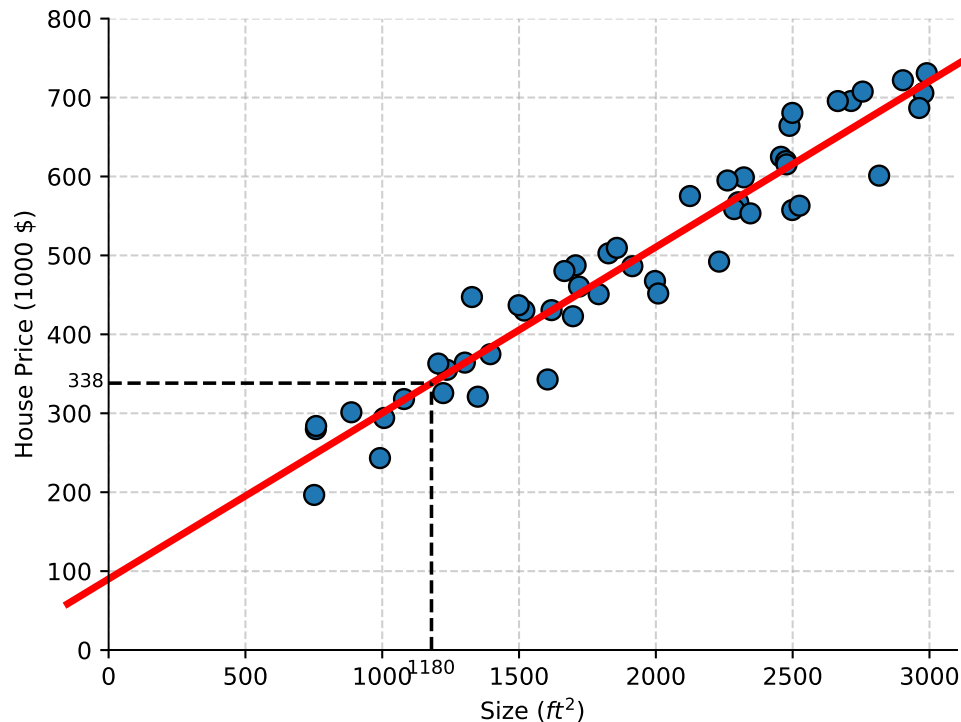
We can also determine
how much of the variance
in price is *explained by*
size.



Simple Linear Regression

Slope is positive →
Positive correlation
between price and size
(i.e. price tends to grow
with size).

We can also determine
how much of the variance
in price is *explained by*
size.

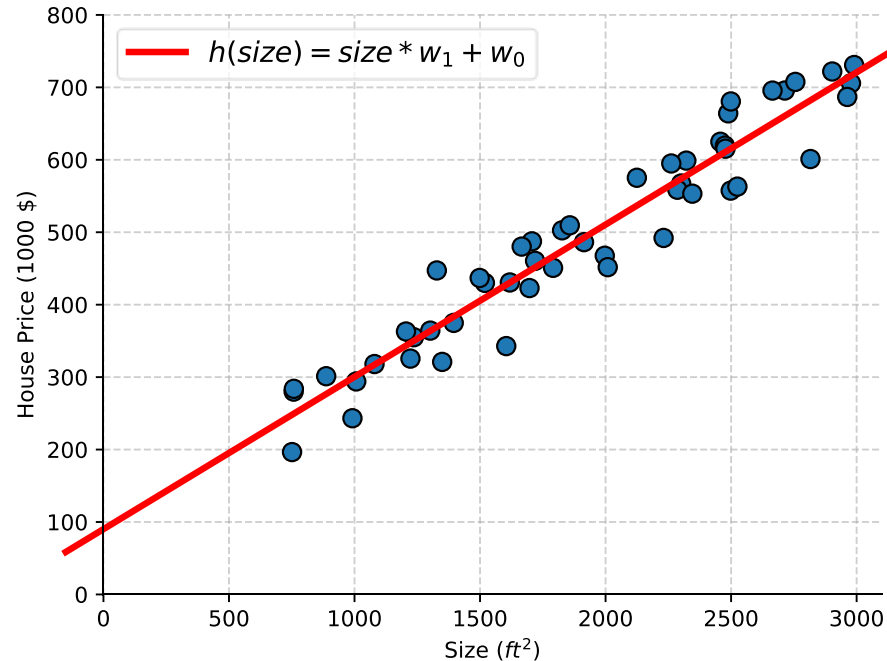


We find a straight *line*
which *best fits* the data.

We can use the model
for forecasting →
The model *predicts*
that a 1220 ft^2 house
will cost \$347K.

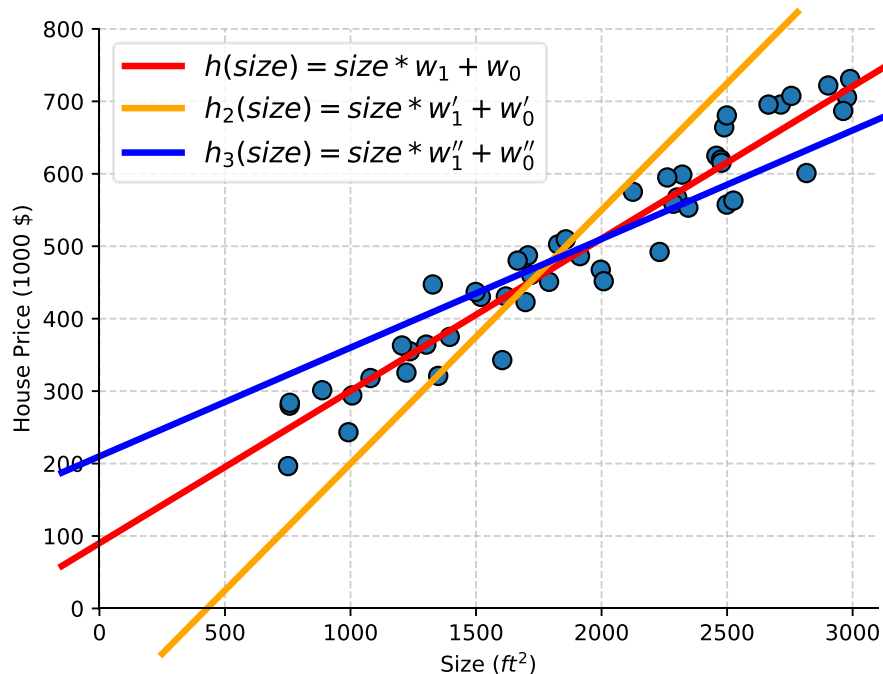
Linear Model

- The relationship is modeled with a *linear function*:



Linear Model

- The relationship is modeled with a *linear function*:



There are many linear functions to choose from.

We need a way of choosing the *best* w_0 and w_1 .

Notation

- $x \in \mathbb{R}$ is the independent variable (i.e. size)
- $y \in \mathbb{R}$ is the dependent variable (i.e. price)
- $h : \mathbb{R} \rightarrow \mathbb{R}$ is the hypothesis, which has parameters w_0 and w_1
- $\hat{y} = h(x)$ is the predicted value.

Notation

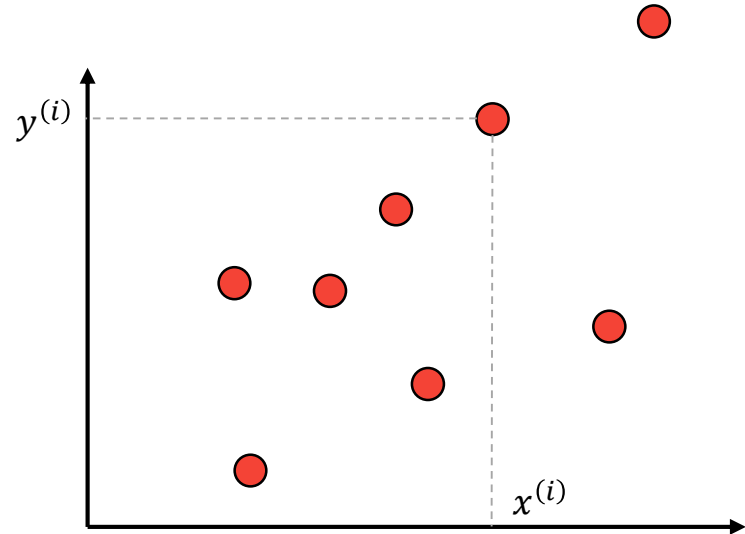
- $x \in \mathbb{R}$ is the independent variable (i.e. size)
- $y \in \mathbb{R}$ is the dependent variable (i.e. price)
- $h : \mathbb{R} \rightarrow \mathbb{R}$ is the hypothesis, which has parameters w_0 and w_1
- $\hat{y} = h(x)$ is the predicted value.
- Simple Linear Regression:

$$\hat{y} = h(x) = w_0 + w_1 x$$

- We need to find w_0 and w_1 such that \hat{y} is as close to y as possible.

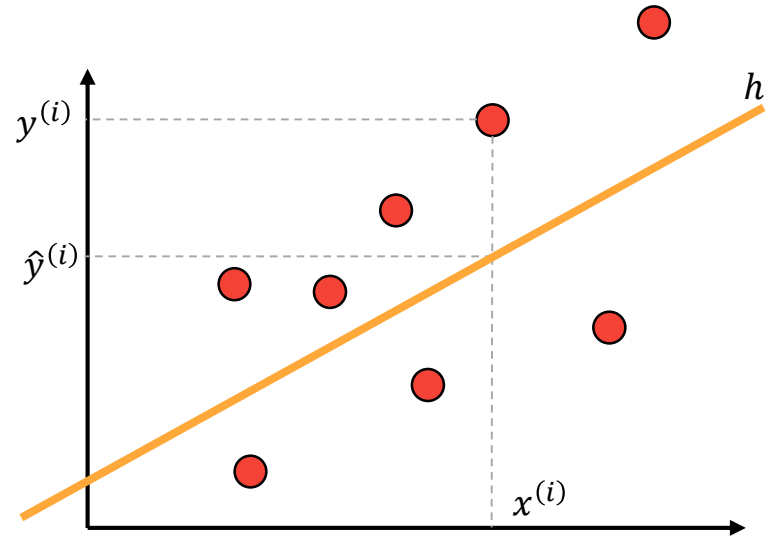
Least Squares Method

- Given $E = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$, with $x^{(i)}, y^{(i)} \in \mathbb{R}$



Least Squares Method

- Given $E = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$, with $x^{(i)}, y^{(i)} \in \mathbb{R}$

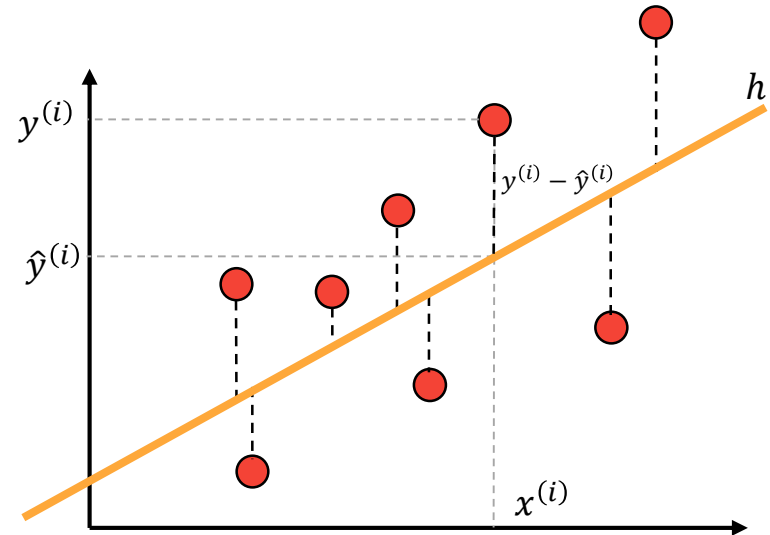


Least Squares Method

- Given $E = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$, with $x^{(i)}, y^{(i)} \in \mathbb{R}$

$$\mathcal{L}_E = \sum_i (y^{(i)} - \hat{y}^{(i)})^2 = \sum_i (y^{(i)} - w_1 x^{(i)} - w_0)^2$$

Squared loss



Least Squares Method

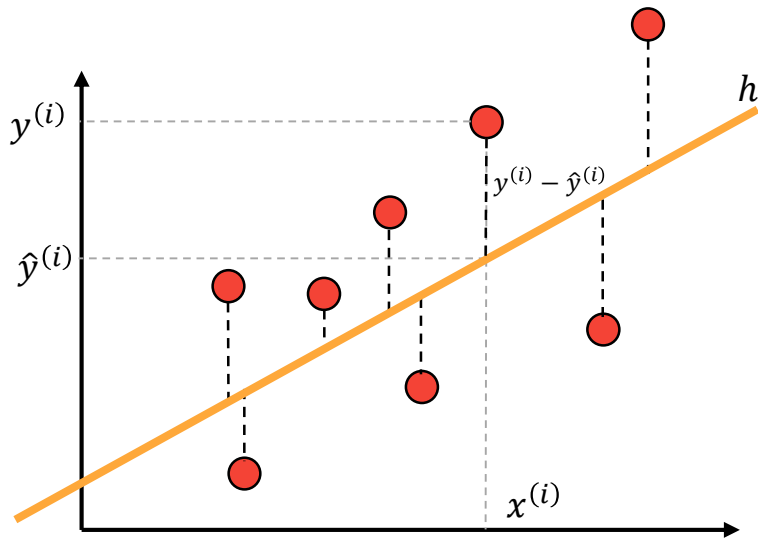
- Given $E = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$, with $x^{(i)}, y^{(i)} \in \mathbb{R}$

$$\mathcal{L}_E = \sum_i (y^{(i)} - \hat{y}^{(i)})^2 = \sum_i (y^{(i)} - w_1 x^{(i)} - w_0)^2$$

Squared loss

- Minimize \mathcal{L}_E w.r.t. w_0, w_1 :

$$\frac{\partial \mathcal{L}_E}{\partial w_0} = 0, \quad \frac{\partial \mathcal{L}_E}{\partial w_1} = 0$$



Least Squares Method

- Given $E = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$, with $x^{(i)}, y^{(i)} \in \mathbb{R}$

$$\mathcal{L}_E = \sum_i (y^{(i)} - \hat{y}^{(i)})^2 = \sum_i (y^{(i)} - w_1 x^{(i)} - w_0)^2$$

Squared loss

- Minimize \mathcal{L}_E w.r.t. w_0, w_1 :

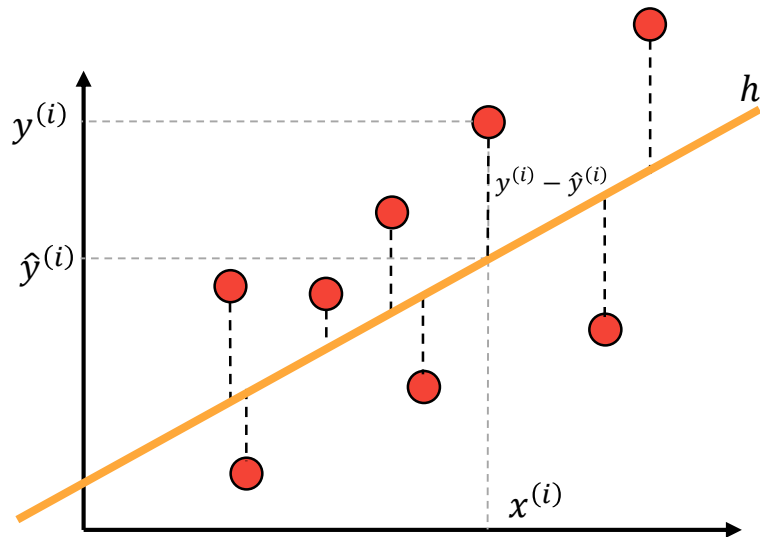
$$\frac{\partial \mathcal{L}_E}{\partial w_0} = 0, \quad \frac{\partial \mathcal{L}_E}{\partial w_1} = 0$$

$$w_1 =$$



Let's do some math...

$$w_0 =$$



Least Squares Derivation

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2$$

To simplify notation
let's assume:

$$y \stackrel{\text{not}}{=} y^{(i)}$$

$$x \stackrel{\text{not}}{=} x^{(i)}$$

Least Squares Derivation

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1x - w_0)^2 = \sum (y^2 + w_1^2x^2 + w_0^2 - 2w_1xy - 2w_0y + 2w_0w_1x)$$

To simplify notation
let's assume:

$y = y^{(i)}$

$x = x^{(i)}$

Least Squares Derivation

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1x - w_0)^2 = \sum (y^2 + w_1^2x^2 + w_0^2 - 2w_1xy - 2w_0y + 2w_0w_1x)$$

$$\frac{\partial \mathcal{L}_E}{\partial w_0} =$$

To simplify notation
let's assume:

$$y \overset{\text{not}}{=} y^{(i)}$$

$$x \overset{\text{not}}{=} x^{(i)}$$

Least Squares Derivation

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1x - w_0)^2 = \sum (y^2 + w_1^2x^2 + w_0^2 - 2w_1xy - 2w_0y + 2w_0w_1x)$$

$$\frac{\partial \mathcal{L}_E}{\partial w_0} = \sum (2w_0 - 2y + 2w_1x)$$

To simplify notation
let's assume:

$$y = y^{(i)}$$

$$x = x^{(i)}$$

Least Squares Derivation

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

$$\frac{\partial \mathcal{L}_E}{\partial w_0} = \sum (2w_0 - 2y + 2w_1 x) = 2 \left(mw_0 - \sum y + w_1 \sum x \right) = 0$$

To simplify notation
let's assume:

$$y = y^{(i)}$$

$$x = x^{(i)}$$

Least Squares Derivation

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

$$w_0 = \frac{1}{m} \left(\sum y - w_1 \sum x \right)$$

To simplify notation
let's assume:

$$y \stackrel{\text{not}}{=} y^{(i)}$$

$$x \stackrel{\text{not}}{=} x^{(i)}$$

Least Squares Derivation

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

$$w_0 = \frac{1}{m} \left(\sum y - w_1 \sum x \right)$$

$$\frac{\partial \mathcal{L}_E}{\partial w_1} =$$

To simplify notation
let's assume:

$$y = y^{(i)}$$

$$x = x^{(i)}$$

Least Squares Derivation

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

$$w_0 = \frac{1}{m} \left(\sum y - w_1 \sum x \right)$$

$$\frac{\partial \mathcal{L}_E}{\partial w_1} = \sum (2w_1 x^2 - 2xy + 2w_0 x)$$

To simplify notation
let's assume:

$$y = y^{(i)}$$

$$x = x^{(i)}$$

Least Squares Derivation

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

$$w_0 = \frac{1}{m} \left(\sum y - w_1 \sum x \right)$$

$$\frac{\partial \mathcal{L}_E}{\partial w_1} = \sum (2w_1 x^2 - 2xy + 2w_0 x) = 2 \left(w_1 \sum x^2 - \sum xy + w_0 \sum x \right) = 0$$

To simplify notation
let's assume:


$$y = y^{(i)}$$

$$x = x^{(i)}$$

Least Squares Derivation

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

$$w_0 = \frac{1}{m} \left(\sum y - w_1 \sum x \right)$$

$$\frac{\partial \mathcal{L}_E}{\partial w_1} = \sum (2w_1 x^2 - 2xy + 2w_0 x) = 2 \left(w_1 \sum x^2 - \sum xy + w_0 \sum x \right) = 0$$


To simplify notation
let's assume:

$$y \stackrel{\text{not}}{=} y^{(i)}$$

$$x \stackrel{\text{not}}{=} x^{(i)}$$

Least Squares Derivation

To simplify notation
let's assume:

$y = y^{(i)}$

$x = x^{(i)}$

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

$$w_0 = \frac{1}{m} \left(\sum y - w_1 \sum x \right)$$

$$\frac{\partial \mathcal{L}_E}{\partial w_1} = \sum (2w_1 x^2 - 2xy + 2w_0 x) = 2 \left(w_1 \sum x^2 - \sum xy + \frac{1}{m} \left(\sum y - w_1 \sum x \right) \sum x \right) = 0$$

Least Squares Derivation

To simplify notation
let's assume:

$$y \stackrel{\text{not}}{=} y^{(i)}$$

$$x \stackrel{\text{not}}{=} x^{(i)}$$

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

$$w_0 = \frac{1}{m} \left(\sum y - w_1 \sum x \right)$$

$$\frac{\partial \mathcal{L}_E}{\partial w_1} = \sum (2w_1 x^2 - 2xy + 2w_0 x) = 2 \left(w_1 \sum x^2 - \sum xy + \frac{1}{m} \left(\sum y - w_1 \sum x \right) \sum x \right) = 0$$

$$w_1 \sum x^2 - w_1 \frac{1}{m} \left(\sum x \right)^2 - \sum xy + \frac{1}{m} \sum x \sum y = 0$$

Least Squares Derivation

To simplify notation
let's assume:

$$y \stackrel{\text{not}}{=} y^{(i)}$$

$$x \stackrel{\text{not}}{=} x^{(i)}$$

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

$$w_0 = \frac{1}{m} \left(\sum y - w_1 \sum x \right)$$

$$\frac{\partial \mathcal{L}_E}{\partial w_1} = \sum (2w_1 x^2 - 2xy + 2w_0 x) = 2 \left(w_1 \sum x^2 - \sum xy + \frac{1}{m} \left(\sum y - w_1 \sum x \right) \sum x \right) = 0$$

$$w_1 \sum x^2 - w_1 \frac{1}{m} \left(\sum x \right)^2 - \sum xy + \frac{1}{m} \sum x \sum y = 0$$

$$\Rightarrow w_1 = \frac{\sum xy - \frac{1}{m} \sum x \sum y}{\sum x^2 - \frac{1}{m} (\sum x)^2}$$

Least Squares Derivation

To simplify notation
let's assume:

$y = y^{(i)}$

$x = x^{(i)}$

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

$$w_0 = \frac{1}{m} \left(\sum y - w_1 \sum x \right)$$

$$\frac{\partial \mathcal{L}_E}{\partial w_1} = \sum (2w_1 x^2 - 2xy + 2w_0 x) = 2 \left(w_1 \sum x^2 - \sum xy + \frac{1}{m} \left(\sum y - w_1 \sum x \right) \sum x \right) = 0$$

$$w_1 \sum x^2 - w_1 \frac{1}{m} \left(\sum x \right)^2 - \sum xy + \frac{1}{m} \sum x \sum y = 0$$

$$\Rightarrow w_1 = \frac{\sum xy - \frac{1}{m} \sum x \sum y}{\sum x^2 - \frac{1}{m} (\sum x)^2} = \frac{m \sum xy - \sum x \sum y}{m \sum x^2 - (\sum x)^2}$$

Least Squares Derivation

To simplify notation
let's assume:

$y = y^{(i)}$
 $x = x^{(i)}$

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

$$w_0 = \frac{1}{m} \left(\sum y - w_1 \sum x \right)$$

$$\frac{\partial \mathcal{L}_E}{\partial w_1} = \sum (2w_1 x^2 - 2xy + 2w_0 x) = 2 \left(w_1 \sum x^2 - \sum xy + \frac{1}{m} \left(\sum y - w_1 \sum x \right) \sum x \right) = 0$$

$$w_1 \sum x^2 - w_1 \frac{1}{m} \left(\sum x \right)^2 - \sum xy + \frac{1}{m} \sum x \sum y = 0$$

$$\Rightarrow w_1 = \frac{\sum xy - \frac{1}{m} \sum x \sum y}{\sum x^2 - \frac{1}{m} (\sum x)^2} = \frac{m \sum xy - \sum x \sum y}{m \sum x^2 - (\sum x)^2} \stackrel{\text{rearranging terms}}{=} \dots = \frac{\sum [(x - \bar{x})(y - \bar{y})]}{\sum (x - \bar{x})^2}$$

$\bar{x} = \frac{1}{m} \sum x^{(i)}$
(the mean)

Least Squares Derivation

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

$$w_0 = \frac{1}{m} \left(\sum y - w_1 \sum x \right)$$

$$w_1 = \frac{m \sum xy - \sum x \sum y}{m \sum x^2 - (\sum x)^2} = \frac{\sum [(x - \bar{x})(y - \bar{y})]}{\sum (x - \bar{x})^2}$$

To simplify notation
let's assume:

$$y = y^{(i)}$$

$$x = x^{(i)}$$

Least Squares Method

- Given $E = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$, with $x^{(i)}, y^{(i)} \in \mathbb{R}$

$$\mathcal{L}_E = \sum_i (y^{(i)} - \hat{y}^{(i)})^2 = \sum_i (y^{(i)} - w_1 x^{(i)} - w_0)^2$$

- Minimize \mathcal{L}_E w.r.t. w_0, w_1 :

Squared loss

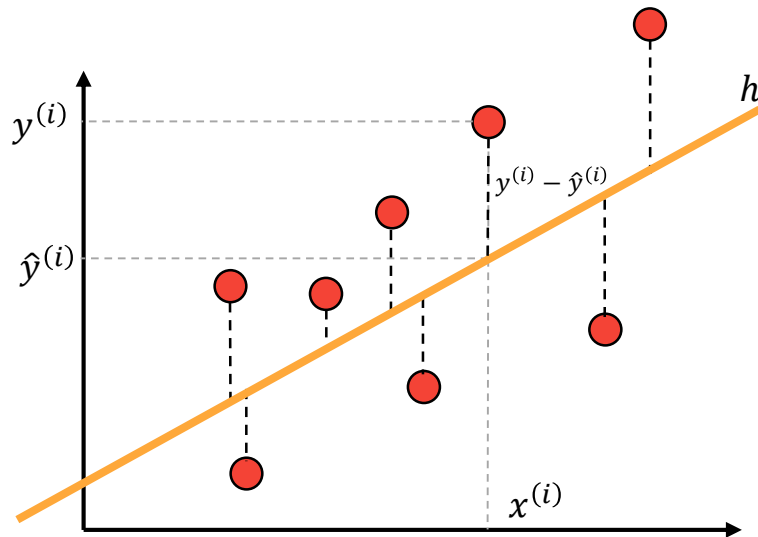
$$\frac{\partial \mathcal{L}_E}{\partial w_0} = 0, \quad \frac{\partial \mathcal{L}_E}{\partial w_1} = 0$$

$$w_1 = \frac{\sum_i [(x^{(i)} - \bar{x})(y^{(i)} - \bar{y})]}{\sum_i (x^{(i)} - \bar{x})^2}$$

\Rightarrow

$$w_0 = \frac{1}{m} \left(\sum_i y^{(i)} - w_1 \sum_i x^{(i)} \right)$$

$$\mathcal{L}_E \stackrel{\text{not}}{=} \sum_{(x,y) \in E} \mathcal{L}(y, h(x))$$



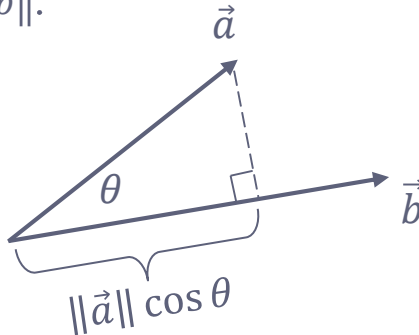
Multiple Linear Regression

Refresher – Dot Product

- An operation which takes two vectors and returns a **scalar** value.
- Also called the *scalar product* or *inner product*.

$$\vec{a} \cdot \vec{b} = \langle \vec{a}, \vec{b} \rangle = \|\vec{a}\| \|\vec{b}\| \cos \theta = \sum_i a_i b_i$$

- It can be interpreted as a **similarity measure** between vectors.
- If we write it as $(\|\vec{a}\| \cos \theta) \|\vec{b}\|$, it can be viewed as the **length of the projection** of \vec{a} on \vec{b} , measured in units of length $\|\vec{b}\|$.



Multiple Regression

- $\vec{x} \in \mathbb{R}^n = [x_1 \quad x_2 \quad \dots \quad x_n],$
- $\vec{w} \in \mathbb{R}^n = [w_1 \quad w_2 \quad \dots \quad w_n], w_0 \in \mathbb{R}$

Multiple Regression

- $\vec{x} \in \mathbb{R}^n = [x_1 \quad x_2 \quad \dots \quad x_n],$
- $\vec{w} \in \mathbb{R}^n = [w_1 \quad w_2 \quad \dots \quad w_n], w_0 \in \mathbb{R}$

$$\hat{y} = w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n = w_0 + \langle \vec{w}, \vec{x} \rangle$$

Multiple Regression

- $\vec{x} \in \mathbb{R}^n = [x_1 \ x_2 \ \dots \ x_n]$,
- $\vec{w} \in \mathbb{R}^n = [w_1 \ w_2 \ \dots \ w_n], w_0 \in \mathbb{R}$

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + \dots + w_n x_n = w_0 + \langle \vec{w}, \vec{x} \rangle$$

- Common mathematical trick is to get rid of the intercept by considering:

$$\vec{x} \in \mathbb{R}^{n+1} = [1 \ x_1 \ x_2 \ \dots \ x_n]$$

$$\vec{w} \in \mathbb{R}^{n+1} = [w_0 \ w_1 \ w_2 \ \dots \ w_n]$$

\Rightarrow

$$\hat{y} = \langle \vec{w}, \vec{x} \rangle$$

Multiple Regression

- Given $E = \{(\vec{x}^{(1)}, y^{(1)}), (\vec{x}^{(2)}, y^{(2)}), \dots, (\vec{x}^{(m)}, y^{(m)})\}$, $\vec{x}^{(i)} \in \mathbb{R}^n, y^{(i)} \in \mathbb{R}$

Multiple Regression

- Given $E = \{(\vec{x}^{(1)}, y^{(1)}), (\vec{x}^{(2)}, y^{(2)}), \dots, (\vec{x}^{(m)}, y^{(m)})\}$, $\vec{x}^{(i)} \in \mathbb{R}^n, y^{(i)} \in \mathbb{R}$
$$\hat{y}^{(i)} = \langle \vec{w}, \vec{x}^{(i)} \rangle = w_0 + w_1 x_1^{(i)} + w_2 x_2^{(i)} + \dots + w_n x_n^{(i)}$$

Multiple Regression

- Given $E = \{(\vec{x}^{(1)}, y^{(1)}), (\vec{x}^{(2)}, y^{(2)}), \dots, (\vec{x}^{(m)}, y^{(m)})\}$, $\vec{x}^{(i)} \in \mathbb{R}^n, y^{(i)} \in \mathbb{R}$

$$\hat{y}^{(i)} = \langle \vec{w}, \vec{x}^{(i)} \rangle = w_0 + w_1 x_1^{(i)} + w_2 x_2^{(i)} + \dots + w_n x_n^{(i)}$$

- We can use matrix multiplication to compute the predictions for all samples:

$$\begin{pmatrix} \hat{y}^{(1)} \\ \hat{y}^{(2)} \\ \vdots \\ \hat{y}^{(m)} \end{pmatrix} = \begin{pmatrix} 1 & x_1^{(1)} & \dots & x_n^{(1)} \\ 1 & x_1^{(2)} & \dots & x_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(m)} & \dots & x_n^{(m)} \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{pmatrix} \Rightarrow \hat{\mathbf{Y}} = \mathbf{X}\mathbf{w}$$

Multiple Regression

- Given $E = \{(\vec{x}^{(1)}, y^{(1)}), (\vec{x}^{(2)}, y^{(2)}), \dots, (\vec{x}^{(m)}, y^{(m)})\}$, $\vec{x}^{(i)} \in \mathbb{R}^n, y^{(i)} \in \mathbb{R}$

$$\hat{y}^{(i)} = \langle \vec{w}, \vec{x}^{(i)} \rangle = w_0 + w_1 x_1^{(i)} + w_2 x_2^{(i)} + \dots + w_n x_n^{(i)}$$

- We can use matrix multiplication to compute the predictions for all samples:

$$\begin{pmatrix} \hat{y}^{(1)} \\ \hat{y}^{(2)} \\ \vdots \\ \hat{y}^{(m)} \end{pmatrix} = \begin{pmatrix} 1 & x_1^{(1)} & \dots & x_n^{(1)} \\ 1 & x_1^{(2)} & \dots & x_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(m)} & \dots & x_n^{(m)} \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{pmatrix} \Rightarrow \hat{\mathbf{Y}} = \mathbf{X}\mathbf{w}$$

$$\mathcal{L}_E = \sum_{i=1}^m (y^{(i)} - \hat{y}^{(i)})^2$$

Multiple Regression

- Given $E = \{(\vec{x}^{(1)}, y^{(1)}), (\vec{x}^{(2)}, y^{(2)}), \dots, (\vec{x}^{(m)}, y^{(m)})\}$, $\vec{x}^{(i)} \in \mathbb{R}^n, y^{(i)} \in \mathbb{R}$

$$\hat{y}^{(i)} = \langle \vec{w}, \vec{x}^{(i)} \rangle = w_0 + w_1 x_1^{(i)} + w_2 x_2^{(i)} + \dots + w_n x_n^{(i)}$$

- We can use matrix multiplication to compute the predictions for all samples:

$$\begin{pmatrix} \hat{y}^{(1)} \\ \hat{y}^{(2)} \\ \vdots \\ \hat{y}^{(m)} \end{pmatrix} = \begin{pmatrix} 1 & x_1^{(1)} & \dots & x_n^{(1)} \\ 1 & x_1^{(2)} & \dots & x_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(m)} & \dots & x_n^{(m)} \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{pmatrix} \Rightarrow \hat{\mathbf{Y}} = \mathbf{X}\mathbf{w}$$

$$\mathcal{L}_E = \sum_{i=1}^m (y^{(i)} - \hat{y}^{(i)})^2 = (\mathbf{Y} - \hat{\mathbf{Y}})^T (\mathbf{Y} - \hat{\mathbf{Y}})$$

Multiple Regression

- Given $E = \{(\vec{x}^{(1)}, y^{(1)}), (\vec{x}^{(2)}, y^{(2)}), \dots, (\vec{x}^{(m)}, y^{(m)})\}$, $\vec{x}^{(i)} \in \mathbb{R}^n, y^{(i)} \in \mathbb{R}$

$$\hat{y}^{(i)} = \langle \vec{w}, \vec{x}^{(i)} \rangle = w_0 + w_1 x_1^{(i)} + w_2 x_2^{(i)} + \dots + w_n x_n^{(i)}$$

- We can use matrix multiplication to compute the predictions for all samples:

$$\begin{pmatrix} \hat{y}^{(1)} \\ \hat{y}^{(2)} \\ \vdots \\ \hat{y}^{(m)} \end{pmatrix} = \begin{pmatrix} 1 & x_1^{(1)} & \dots & x_n^{(1)} \\ 1 & x_1^{(2)} & \dots & x_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(m)} & \dots & x_n^{(m)} \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{pmatrix} \Rightarrow \hat{\mathbf{Y}} = \mathbf{X}\mathbf{w}$$

$$\mathcal{L}_E = \sum_{i=1}^m (y^{(i)} - \hat{y}^{(i)})^2 = (\mathbf{Y} - \hat{\mathbf{Y}})^T (\mathbf{Y} - \hat{\mathbf{Y}}) \stackrel{\text{not}}{=} (\mathbf{Y} - \hat{\mathbf{Y}})^2$$

Multiple Regression

$$\mathcal{L}_E = (Y - \hat{Y})^2$$

Multiple Regression

$$\mathcal{L}_E = (Y - \hat{Y})^2 = (Y - Xw)^2 = (Y - Xw)^T (Y - Xw)$$

Multiple Regression

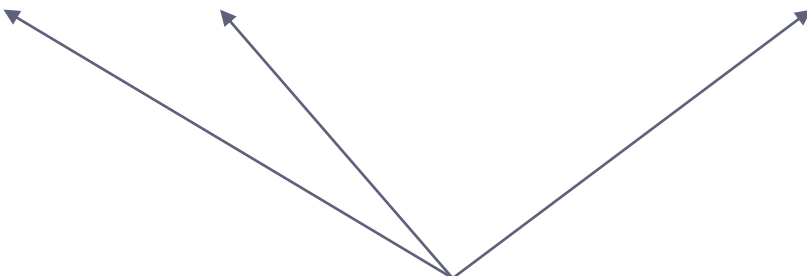
$$\begin{aligned}\mathcal{L}_E &= (Y - \hat{Y})^2 = (Y - Xw)^2 = (Y - Xw)^T (Y - Xw) \\ &= (Y^T - w^T X^T)(Y - Xw)\end{aligned}$$

Multiple Regression

$$\begin{aligned}\mathcal{L}_E &= (Y - \hat{Y})^2 = (Y - Xw)^2 = (Y - Xw)^T(Y - Xw) \\ &= (Y^T - w^T X^T)(Y - Xw) \\ &= Y^T Y - Y^T Xw - w^T X^T Y + w^T X^T Xw\end{aligned}$$

Multiple Regression

$$\begin{aligned}\mathcal{L}_E &= (Y - \hat{Y})^2 = (Y - Xw)^2 = (Y - Xw)^T(Y - Xw) \\ &= (Y^T - w^T X^T)(Y - Xw) \\ &= Y^T Y - Y^T Xw - w^T X^T Y + w^T X^T Xw = Y^T Y - 2w^T X^T Y + w^T X^T Xw\end{aligned}$$

- 
- $Y^T Xw = [\cdot]_{1 \times m} [\cdot]_{m \times n} [\cdot]_{n \times 1} = [\cdot]_{1 \times 1}$ (scalar)
 - $Y^T Xw = (w^T X^T Y)^T$
 - Transpose of a scalar is the same scalar

Multiple Regression

$$\mathcal{L}_E = Y^T Y - 2w^T X^T Y + w^T X^T X w$$

Multiple Regression

$$\mathcal{L}_E = Y^T Y - 2w^T X^T Y + w^T X^T X w$$

- Minimize \mathcal{L}_E with respect to w :

$$\frac{\partial \mathcal{L}_E}{\partial w} = 0$$

Derivative w.r.t. to a vector \Rightarrow
derivative w.r.t. each component.

$$\begin{bmatrix} \frac{\partial \mathcal{L}_E}{\partial w_0} & \frac{\partial \mathcal{L}_E}{\partial w_1} & \dots \end{bmatrix}$$

Multiple Regression

$$\mathcal{L}_E = Y^T Y - 2w^T X^T Y + w^T X^T X w$$

- Minimize \mathcal{L}_E with respect to w :

$$\frac{\partial \mathcal{L}_E}{\partial w} = 0 \Rightarrow -2X^T Y + 2X^T X w = 0$$

Derivative w.r.t. to a vector \Rightarrow
derivative w.r.t. each component.

$$\begin{bmatrix} \frac{\partial \mathcal{L}_E}{\partial w_0} & \frac{\partial \mathcal{L}_E}{\partial w_1} & \dots \end{bmatrix}$$

Multiple Regression

$$\mathcal{L}_E = Y^T Y - 2w^T X^T Y + w^T X^T X w$$

- Minimize \mathcal{L}_E with respect to w :

$$\frac{\partial \mathcal{L}_E}{\partial w} = 0 \Rightarrow -2X^T Y + 2X^T X w = 0 \Rightarrow X^T X w = X^T Y$$

Derivative w.r.t. to a vector \Rightarrow
derivative w.r.t. each component.

$$\begin{bmatrix} \frac{\partial \mathcal{L}_E}{\partial w_0} & \frac{\partial \mathcal{L}_E}{\partial w_1} & \dots \end{bmatrix}$$

Multiple Regression

$$\mathcal{L}_E = Y^T Y - 2w^T X^T Y + w^T X^T X w$$

- Minimize \mathcal{L}_E with respect to w :

$$\frac{\partial \mathcal{L}_E}{\partial w} = 0 \Rightarrow -2X^T Y + 2X^T X w = 0 \Rightarrow X^T X w = X^T Y \Rightarrow$$

$$w = (X^T X)^{-1} X^T Y$$

Derivative w.r.t. to a vector \Rightarrow
derivative w.r.t. each component.

$$\begin{bmatrix} \frac{\partial \mathcal{L}_E}{\partial w_0} & \frac{\partial \mathcal{L}_E}{\partial w_1} & \dots \end{bmatrix}$$

Multiple Regression

$$\mathcal{L}_E = Y^T Y - 2w^T X^T Y + w^T X^T X w$$

- Minimize \mathcal{L}_E with respect to w :

$$\frac{\partial \mathcal{L}_E}{\partial w} = 0 \Rightarrow -2X^T Y + 2X^T X w = 0 \Rightarrow X^T X w = X^T Y \Rightarrow$$

$$w = (X^T X)^{-1} X^T Y$$

- Observations:
 - $X^T X$ needs to be *invertible*.
 - If $X \in \mathbb{R}^{m \times n}$ with $n > m$ (more features than examples), there is a high chance of $X^T X$ being singular.
 - The problem is “*ill-posed*”.

Derivative w.r.t. to a vector \Rightarrow
derivative w.r.t. each component.

$$\begin{bmatrix} \frac{\partial \mathcal{L}_E}{\partial w_0} & \frac{\partial \mathcal{L}_E}{\partial w_1} & \dots \end{bmatrix}$$

Well-posed vs. Ill-posed problems

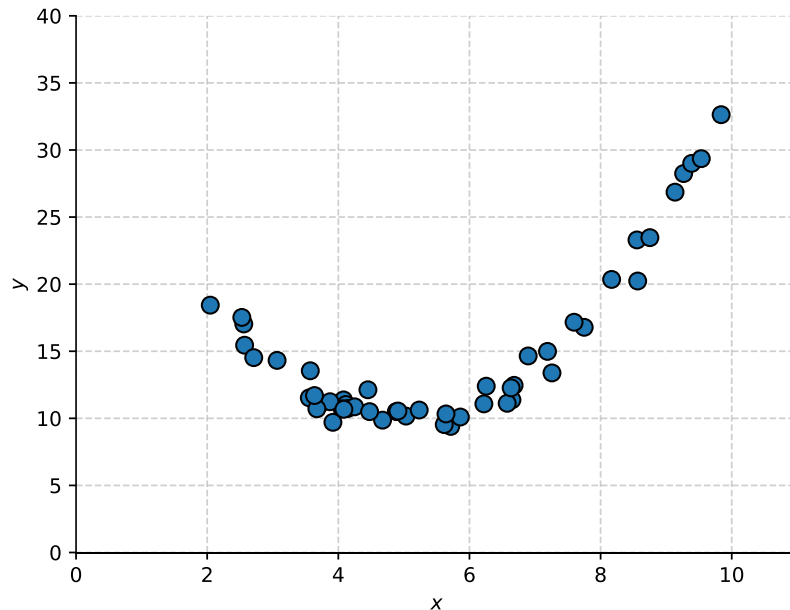
- A problem is **well-posed** if:
 - A solution exists.
 - The solution is unique.
 - The solution's behavior changes continuously with the initial conditions.

Jacques Hadamard, 1902

- If a problem is not well-posed, it is said to be **ill-posed**.
 - This usually implies that additional assumptions need to be taken into consideration for numerical treatment of the problem.
 - This process is called **regularization**
 - Inverse problems are often ill-posed (determining the cause by observing the effects)

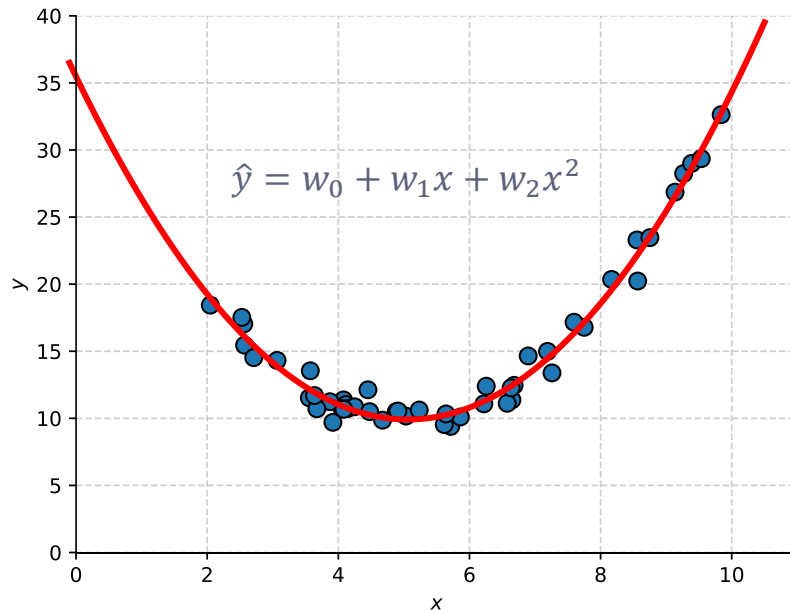
Fitting a polynomial

- Linear regression is *linear in coefficients*, but we can have *non-linear features*.
 - If we use $[1 \ x \ x^2 \ \dots \ x^d]$ as features, we can fit a polynomial of degree d with linear regression



Fitting a polynomial

- Linear regression is *linear in coefficients*, but we can have *non-linear features*.
 - If we use $[1 \ x \ x^2 \ \dots \ x^d]$ as features, we can fit a polynomial of degree d with linear regression



Ridge Regression

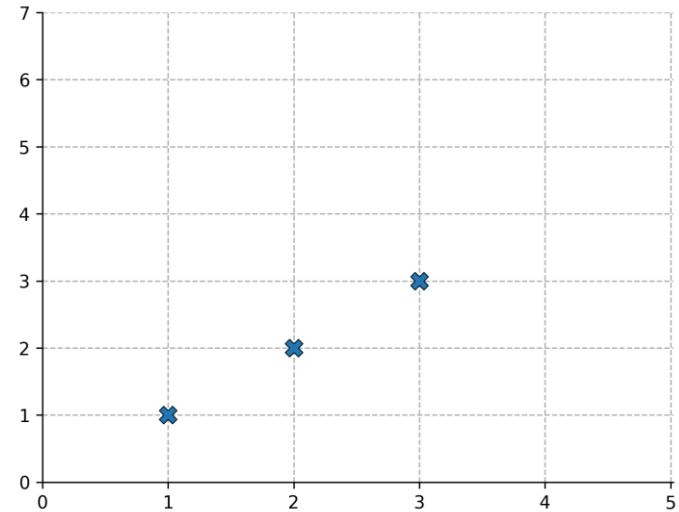
Linear regression with **regularization**

Faculty of Mathematics and Computer Science, University of Bucharest
and
Sparktech Software

Academic Year 2018/2019, 1st Semester

Ill-posed problems

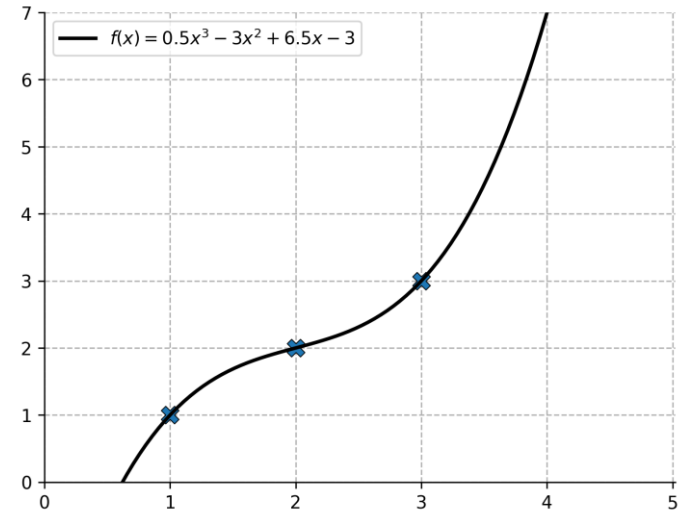
- Let's say we have a function $f: \mathbb{R} \rightarrow \mathbb{R}$
- We are given:
 - $f(1) = 1$
 - $f(2) = 2$
 - $f(3) = 3$
- What is $f(4)$?
 - $f(4) = ?$



Ill-posed problems

- Let's say we have a function $f: \mathbb{R} \rightarrow \mathbb{R}$
- We are given:
 - $f(1) = 1$
 - $f(2) = 2$
 - $f(3) = 3$
- What is $f(4)$?
 - $f(4) = 7$

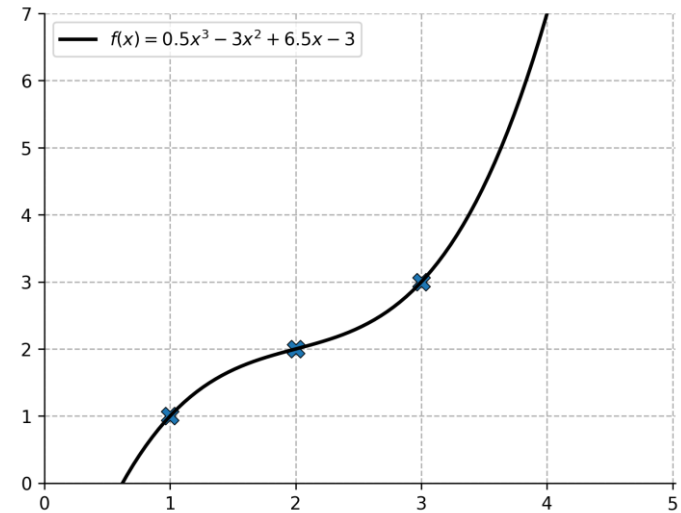
$$f(x) = 0.5x^3 - 3x^2 + 6.5x - 3$$



Ill-posed problems

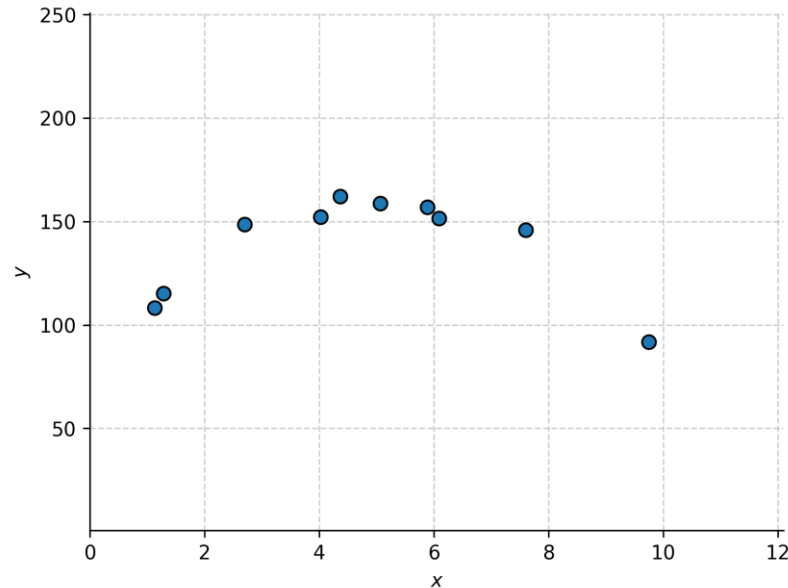
- Let's say we have a function $f: \mathbb{R} \rightarrow \mathbb{R}$
- We are given:
 - $f(1) = 1$
 - $f(2) = 2$
 - $f(3) = 3$
- What is $f(4)$?
 - $f(4) = 7$

$$f(x) = 0.5x^3 - 3x^2 + 6.5x - 3$$



Ill-posed problems

- Lets consider a dataset $X \in \mathbb{R}^{10 \times 10}$ (10 samples with 10 features each)
 - For ease of plotting let's consider the features to be $[1 \ x \ x^2 \ \dots \ x^{10}]$

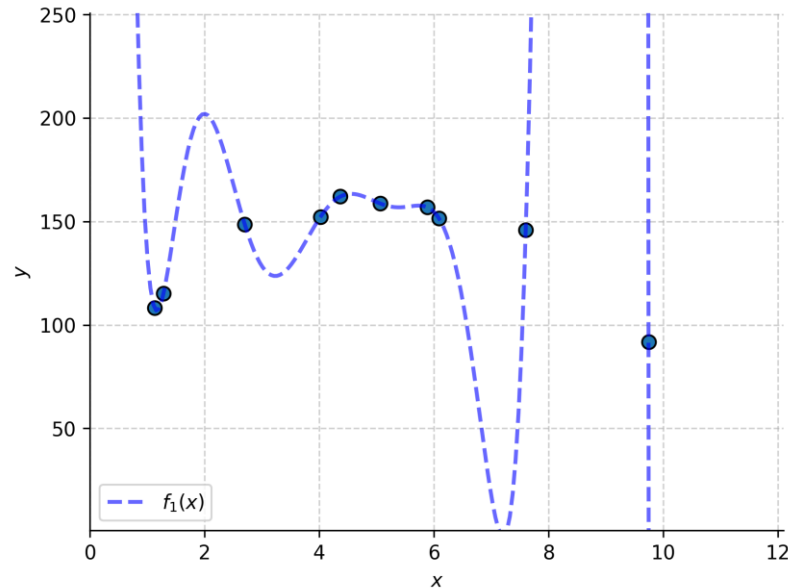


Ill-posed problems

- Lets consider a dataset $X \in \mathbb{R}^{10 \times 10}$ (10 samples with 10 features each)
 - For ease of plotting let's consider the features to be $[1 \ x \ x^2 \ \dots \ x^{10}]$

“Ill-posed” problem +
unconstrained hypothesis
space \rightarrow data is overfitted.

$$\begin{aligned} f_1(x) = & 5.28 * 10^3 \\ & -1.50 * 10^4 x \\ & +1.72 * 10^4 x^2 \\ & -1.01 * 10^4 x^3 + \dots \end{aligned}$$



Ill-posed problems

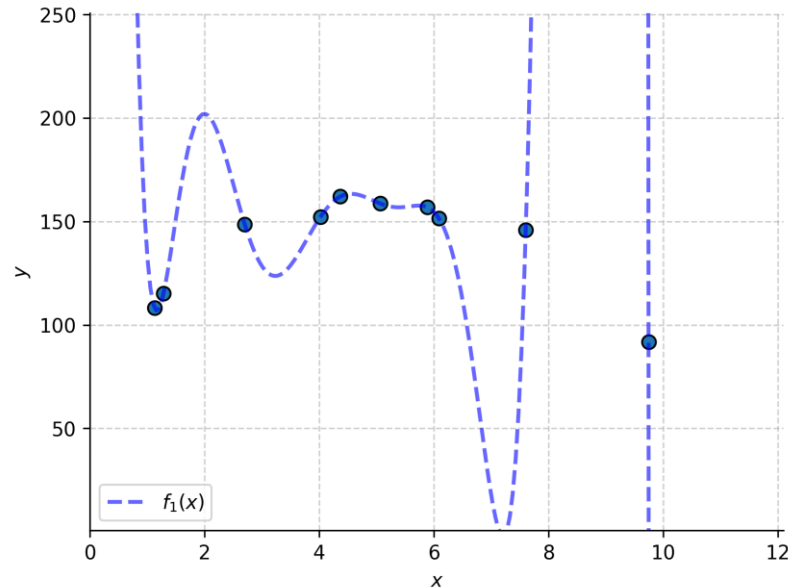
- Let's consider a dataset $X \in \mathbb{R}^{10 \times 10}$ (10 samples with 10 features each)
 - For ease of plotting let's consider the features to be $[1 \ x \ x^2 \ \dots \ x^{10}]$

"Ill-posed" problem +
unconstrained hypothesis
space \rightarrow data is overfitted.

$$f_1(x) = \begin{aligned} &5.28 * 10^3 \\ &- 1.50 * 10^4 x \\ &+ 1.72 * 10^4 x^2 \\ &- 1.01 * 10^4 x^3 + \dots \end{aligned}$$

Large coefficients \rightarrow

- Small changes in input cause large changes in output.



Ill-posed problems

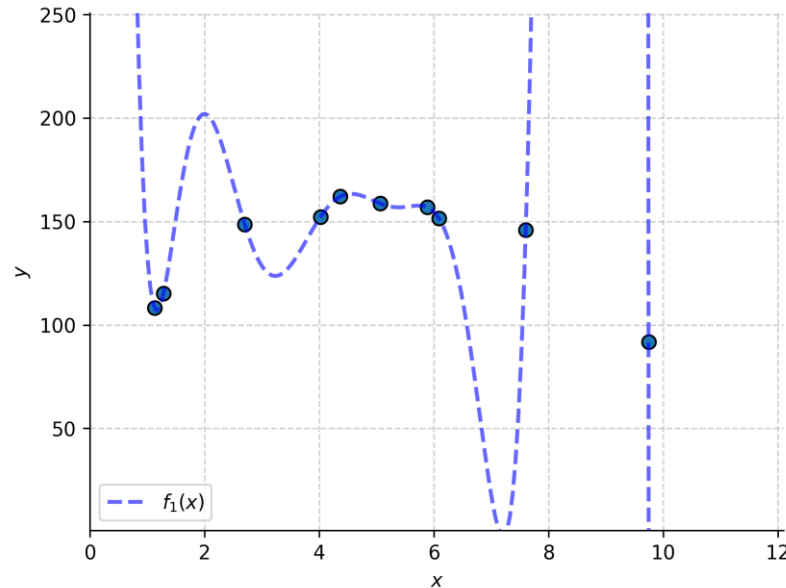
- Lets consider a dataset $X \in \mathbb{R}^{10 \times 10}$ (10 samples with 10 features each)
 - For ease of plotting let's consider the features to be $[1 \ x \ x^2 \ \dots \ x^{10}]$

“Ill-posed” problem +
unconstrained hypothesis
space \rightarrow data is overfitted.

$$f_1(x) = \begin{aligned} &5.28 * 10^3 \\ &- 1.50 * 10^4 x \\ &+ 1.72 * 10^4 x^2 \\ &- 1.01 * 10^4 x^3 + \dots \end{aligned}$$

Large coefficients \rightarrow

- Small changes in input cause large changes in output.



What if we *force*
coefficients to be small?

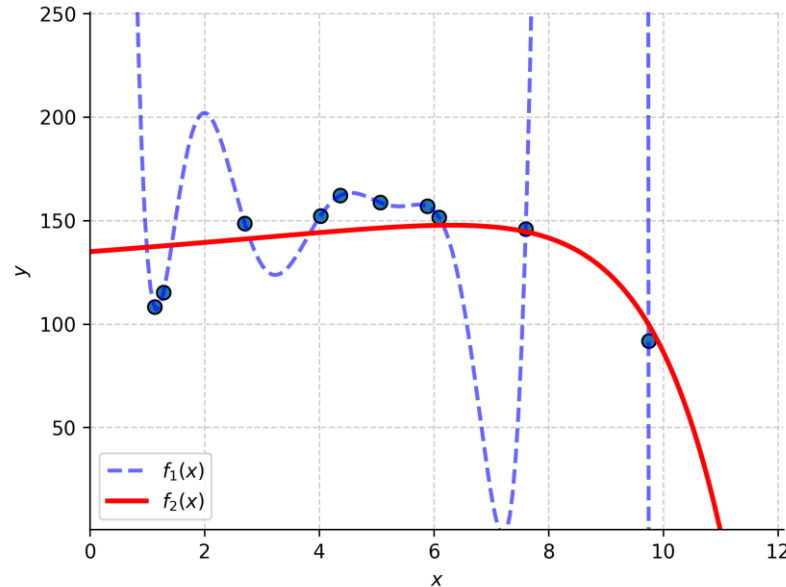
Ill-posed problems

- Lets consider a dataset $X \in \mathbb{R}^{10 \times 10}$ (10 samples with 10 features each)
 - For ease of plotting let's consider the features to be $[1 \ x \ x^2 \ \dots \ x^{10}]$

“Ill-posed” problem +
unconstrained hypothesis
space \rightarrow data is overfitted.

$$f_1(x) = 5.28 * 10^3 - 1.50 * 10^4 x + 1.72 * 10^4 x^2 - 1.01 * 10^4 x^3 + \dots$$

- Large coefficients \rightarrow
- Small changes in input cause large changes in output.



What if we *force*
coefficients to be small?

$$f_2(x) = 1.34 * 10^1 + 2.09 * 10^1 x + 7.11 * 10^{-2} x^2 - 6.24 * 10^{-4} x^3 + \dots$$

- Small coefficients \rightarrow
- Function is much smoother.

Ridge Regression

- We want small coefficients, so we add the norm of the weight vector to the loss:

$$\mathcal{L}_E = \sum_i (y^{(i)} - \hat{y}^{(i)})^2 + \lambda \|\vec{w}\|_2^2$$

Ridge Regression

- We want small coefficients, so we add the norm of the weight vector to the loss:

$$\mathcal{L}_E = \sum_i (y^{(i)} - \hat{y}^{(i)})^2 + \lambda \|\vec{w}\|_2^2$$

L_2 regularization

Ridge Regression

- We want small coefficients, so we add the norm of the weight vector to the loss:

$$\mathcal{L}_E = \sum_i (y^{(i)} - \hat{y}^{(i)})^2 + \lambda \|\vec{w}\|_2^2$$

L_2 regularization

- In matrix format:

$$\mathcal{L}_E = (Y - \hat{Y})^2 + \lambda w^T w$$

Ridge Regression

- We want small coefficients, so we add the norm of the weight vector to the loss:

$$\mathcal{L}_E = \sum_i (y^{(i)} - \hat{y}^{(i)})^2 + \lambda \|\vec{w}\|_2^2$$

L_2 regularization

- In matrix format:

$$\mathcal{L}_E = (Y - \hat{Y})^2 + \lambda w^T w$$

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \Rightarrow$$

$$w = (X^T X + \lambda I)^{-1} X^T Y$$

Ridge Regression

- We want small coefficients, so we add the norm of the weight vector to the loss:

$$\mathcal{L}_E = \sum_i (y^{(i)} - \hat{y}^{(i)})^2 + \lambda \|\vec{w}\|_2^2$$

L_2 regularization

- In matrix format:

$$\mathcal{L}_E = (Y - \hat{Y})^2 + \lambda w^T w$$

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \Rightarrow$$

$$w = (X^T X + \lambda I)^{-1} X^T Y$$

Always invertible.

Other types of regularization

- **Lasso**

- Same as Ridge Regression, but with L_1 regularization:

$$\mathcal{L}_E = \sum_i (y^{(i)} - \hat{y}^{(i)})^2 + \lambda \|\vec{w}\|_1$$

L_1 regularization

- The L_1 penalty encourages some of the weights to be exactly 0, not just small.

Other types of regularization

- **Lasso**

- Same as Ridge Regression, but with L_1 regularization:

$$\mathcal{L}_E = \sum_i (y^{(i)} - \hat{y}^{(i)})^2 + \lambda \|\vec{w}\|_1$$

L_1 regularization

- The L_1 penalty encourages some of the weights to be exactly 0, not just small.

- **Elastic Net**

- Linear combination between Ridge Regression and Lasso Regression:

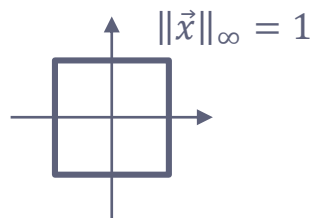
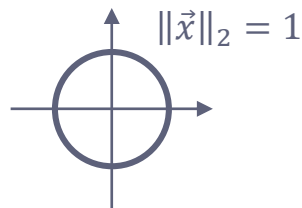
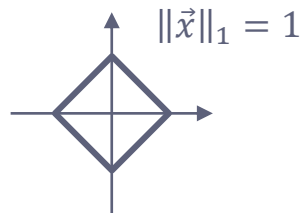
$$\mathcal{L}_E = \sum_i (y^{(i)} - \hat{y}^{(i)})^2 + \lambda_1 \|\vec{w}\|_2^2 + \lambda_2 \|\vec{w}\|_1$$

Other types of regularization

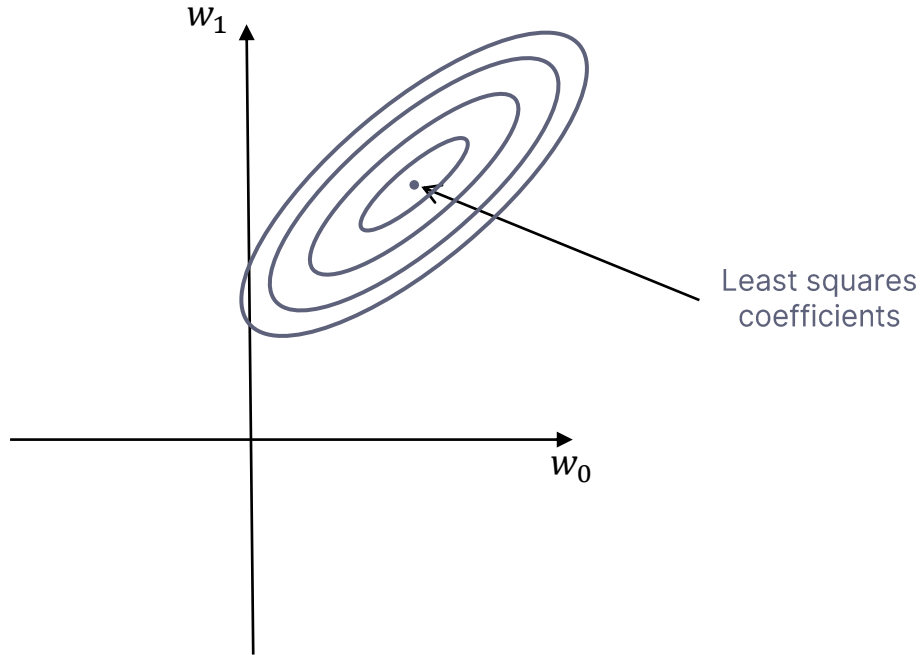
- L_p -norm of a vector $\vec{x} \in \mathbb{R}^n$:

$$\|\vec{x}\|_p = \sqrt[p]{\sum_{i=1}^n |x_i|^p}$$

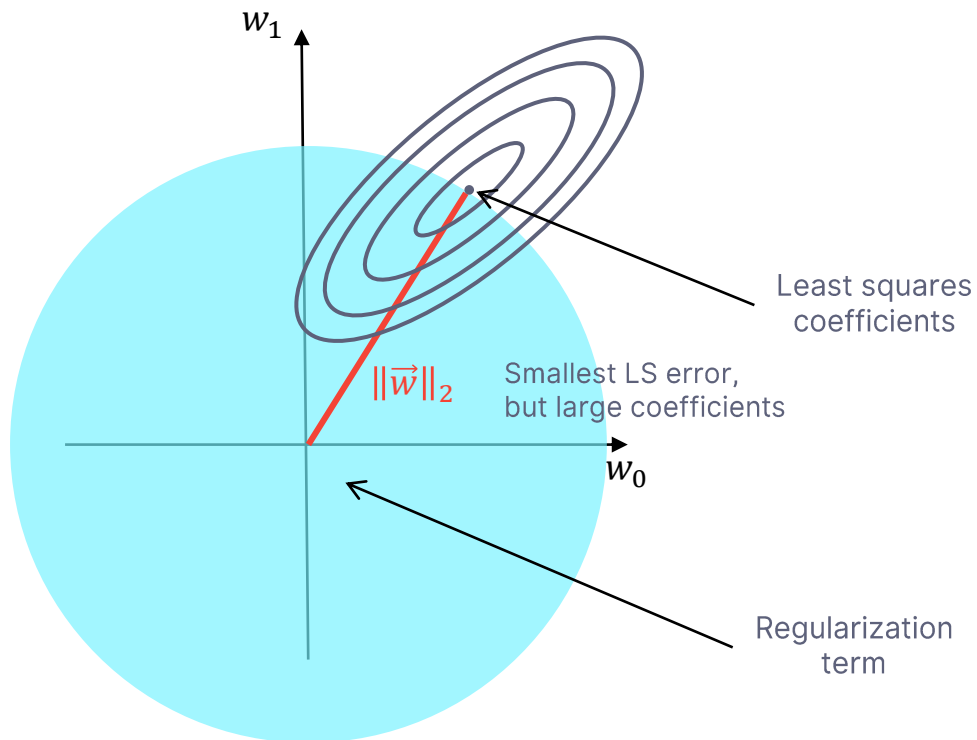
- $p = 1 \Rightarrow$ Manhattan Norm (sum of absolute element values)
- $p = 2 \Rightarrow$ Euclidean Norm
- $p = \infty \Rightarrow$ Maximum Norm (value of the absolute maximum element)



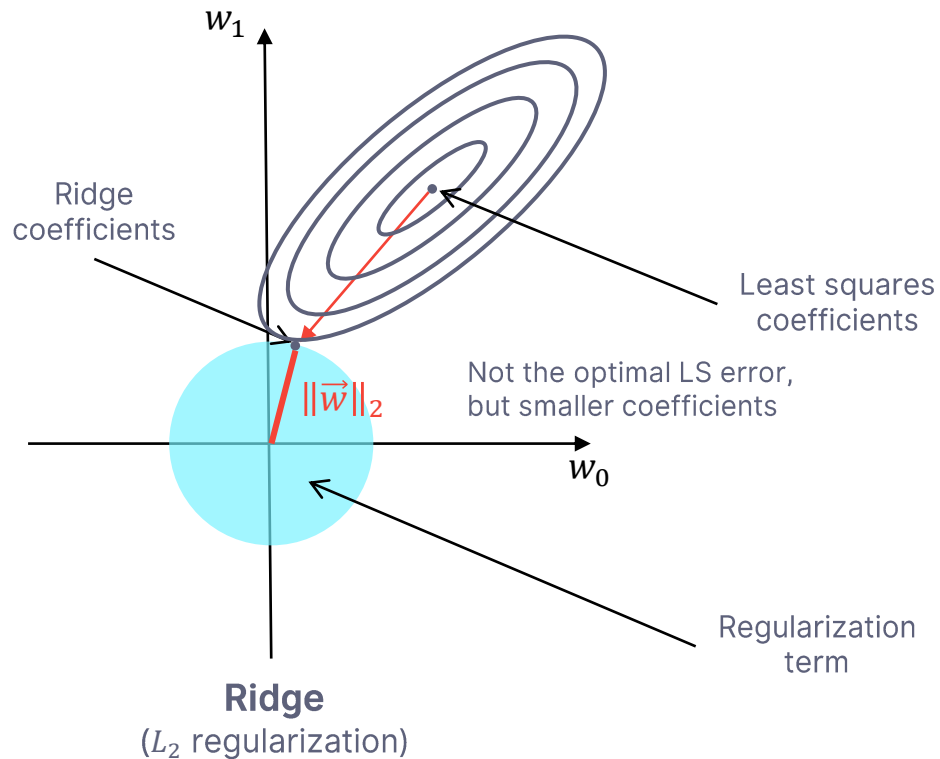
Other types of regularization



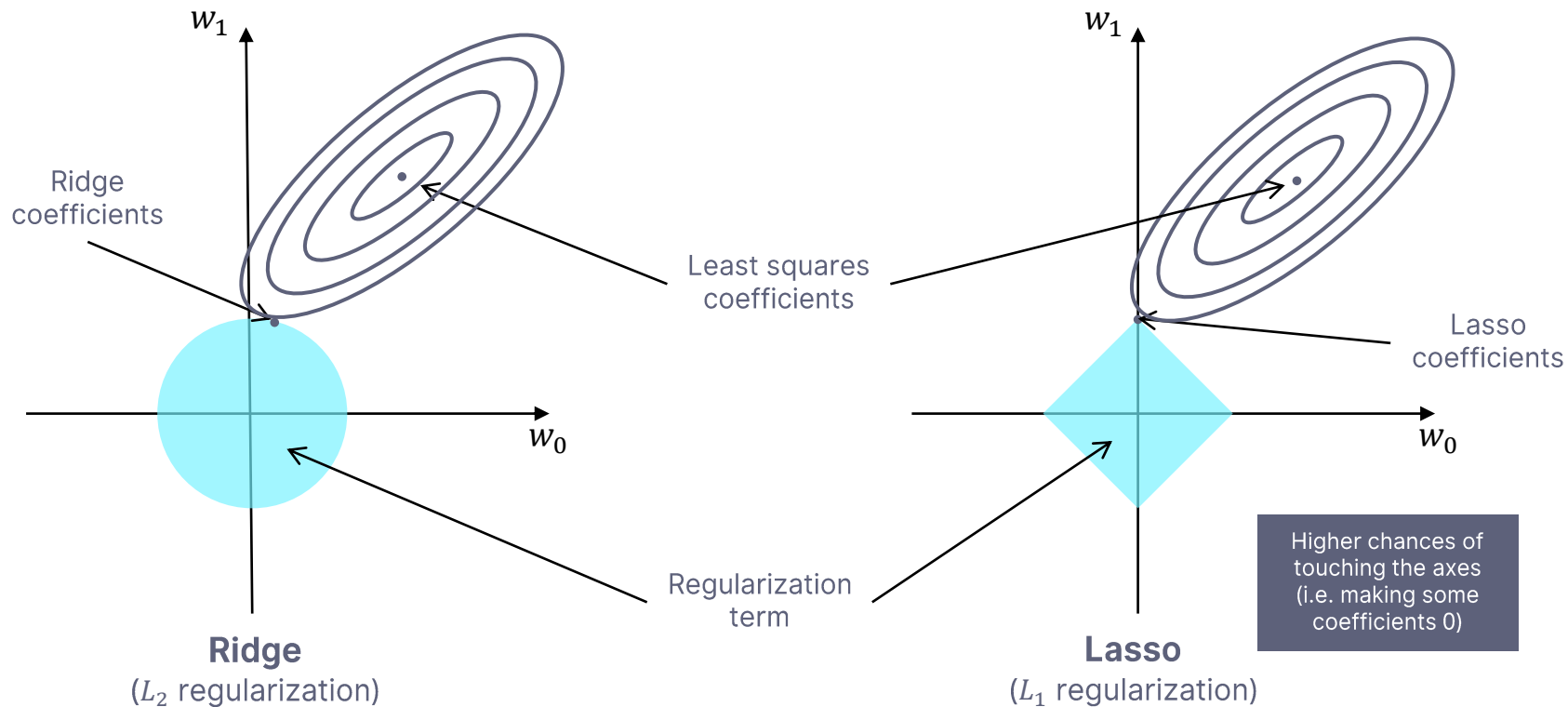
Other types of regularization



Other types of regularization



Other types of regularization



Recap

- **Linear regression**

- Fits a *linear model* on the data:

- $\hat{y} = \langle \vec{x}, \vec{w} \rangle = w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n$
- in matrix form: $\hat{Y} = Xw$

- Minimizing the *squared-error loss* gives the following formula:

- $w = (X^T X)^{-1} X^T Y$

- **Ridge regression**

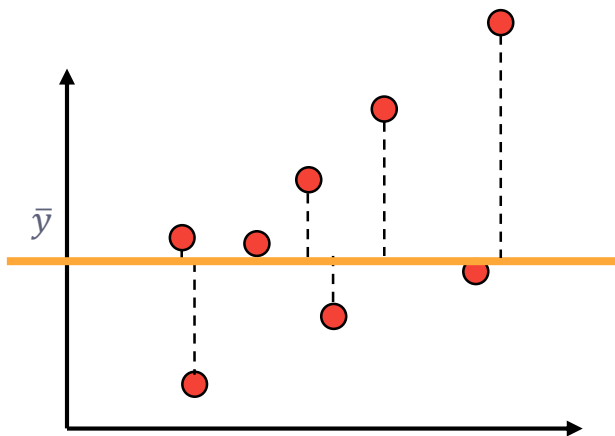
- Forcing small coefficients makes the function smoother and less prone to overfitting.
- Adding L_2 *regularization* (the Euclidean norm of the weight vector) to the loss gives:

- $w = (X^T X + \lambda I)^{-1} X^T Y$

Coefficient of determination

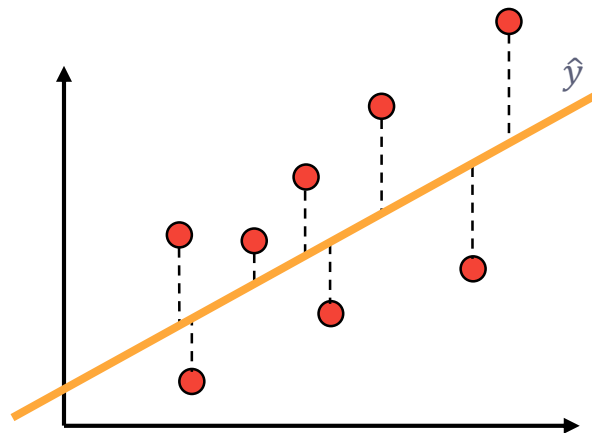
- The coefficient of determination (R^2 , pronounced “R squared”)
 - amount of variance in the dependent variable which is explained by the independent variables.

$$\overline{\mathcal{L}_E} = \sum_i (y^{(i)} - \bar{y})^2$$



$$R^2 \stackrel{\text{def}}{=} \frac{\overline{\mathcal{L}_E} - \mathcal{L}_E(\vec{w})}{\overline{\mathcal{L}_E}}$$

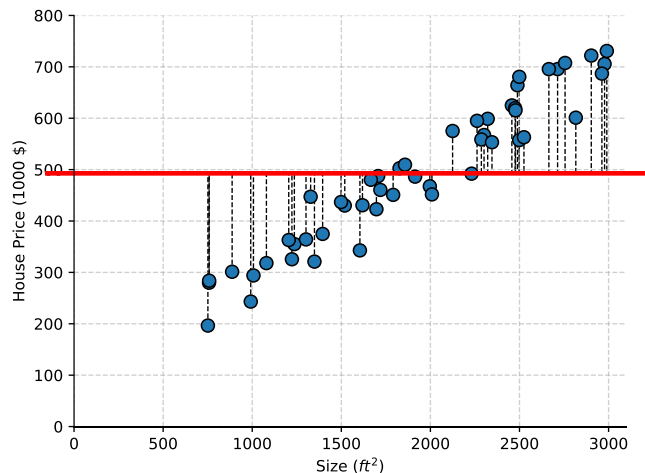
$$\mathcal{L}_E(\vec{w}) = \sum_i (y^{(i)} - \hat{y}^{(i)})^2$$



Coefficient of determination

$$R^2 = \frac{939,000.6 - 73,032.3}{939,000.6} = 0.922$$

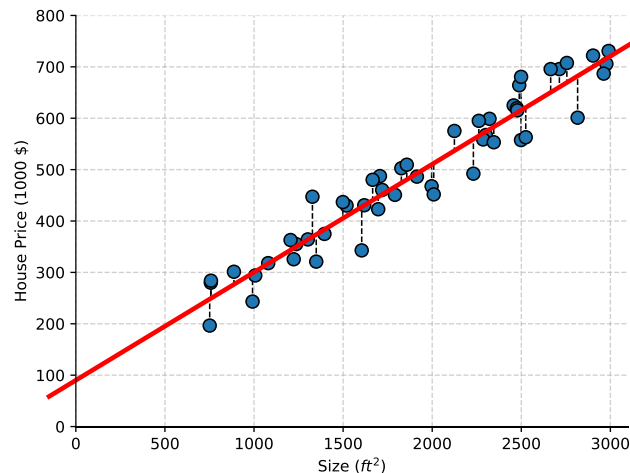
$$\overline{\mathcal{L}_E} = 939,000.6$$



92.2% of house price is explained by size.

BTW: This is a fake dataset!

$$\mathcal{L}_E(\vec{w}) = 73,032.3$$



Linear Regression in Python

```
1  from sklearn.linear_model import LinearRegression, Ridge, Lasso, ElasticNet
2
3  clf = LinearRegression()
4  clf.fit(X, y)
5
6  clf.predict([x]) # prediction for x
7
8  clf.score(X, y) #  $R^2$  coefficient
9  clf._residues # Loss on the training samples  $\mathcal{L}_{\hat{y}}$ 
10 clf.coef_ #  $w_1, w_2 \dots w_n$ 
11 clf.intercept_ #  $w_0$ 
12
13 clf = Ridge(alpha = 10) # alpha is the regularization strength  $\lambda$ 
```


Summary

- **Linear Regression** uses a *linear function* to establish a relationship between a *dependent variable* and a number of *independent variables*.
- The parameters of the model are obtained by minimizing the **squared-error loss**.
- The fitted model can be used to determine the *correlation* between features and label.
- The fitted model can also be used to make *predictions* on future data.
- **Ridge Regression** adds *regularization* in order to deal with *ill-posed problems*.

Keywords

Linear Regression

Dependent / Independent Variable

Multivariate

Linear Function

Squared-error Loss

Ill-posed Problem

Regularization

Ridge Regression

Lasso

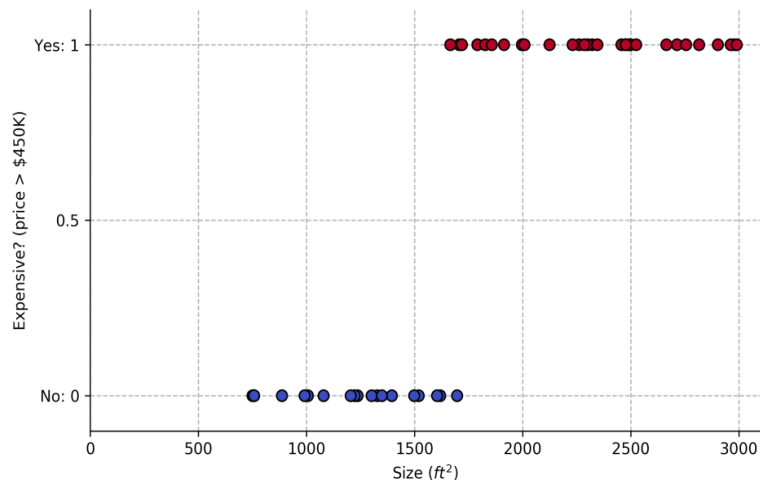
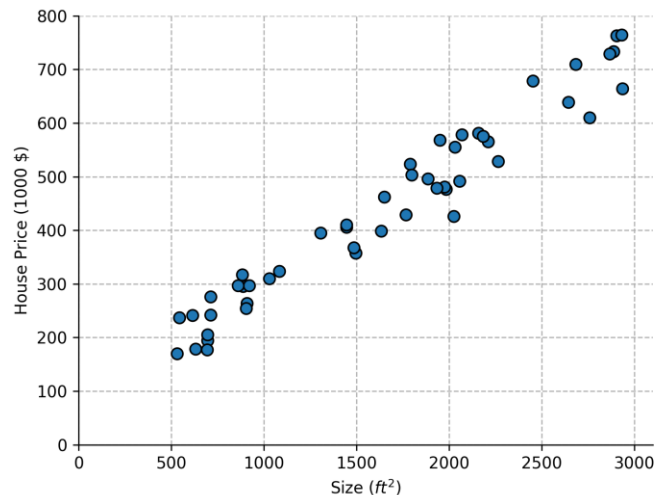
Elastic Net

Correlation

R^2 (R squared)

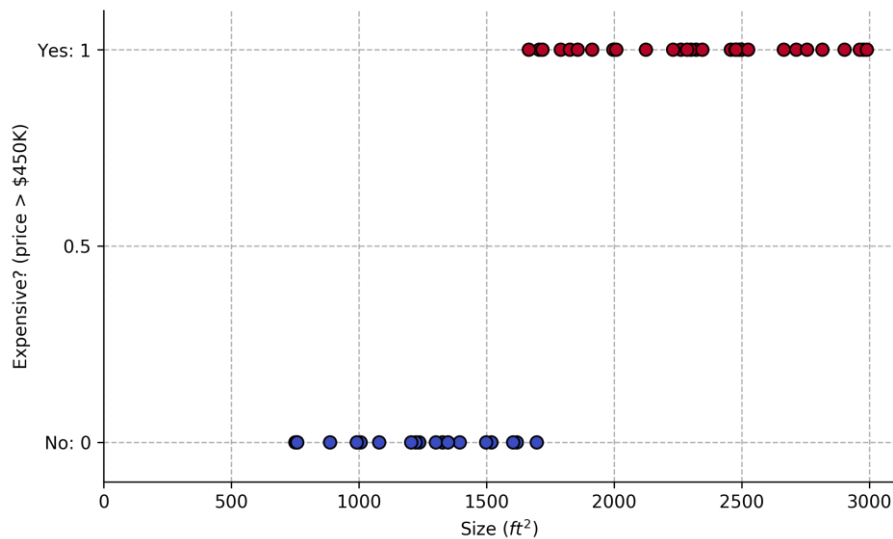
Regression vs. Classification

- What if we only knew and wanted to predict whether a house is expensive or not?
 - No information about the actual price → Binary classification problem.



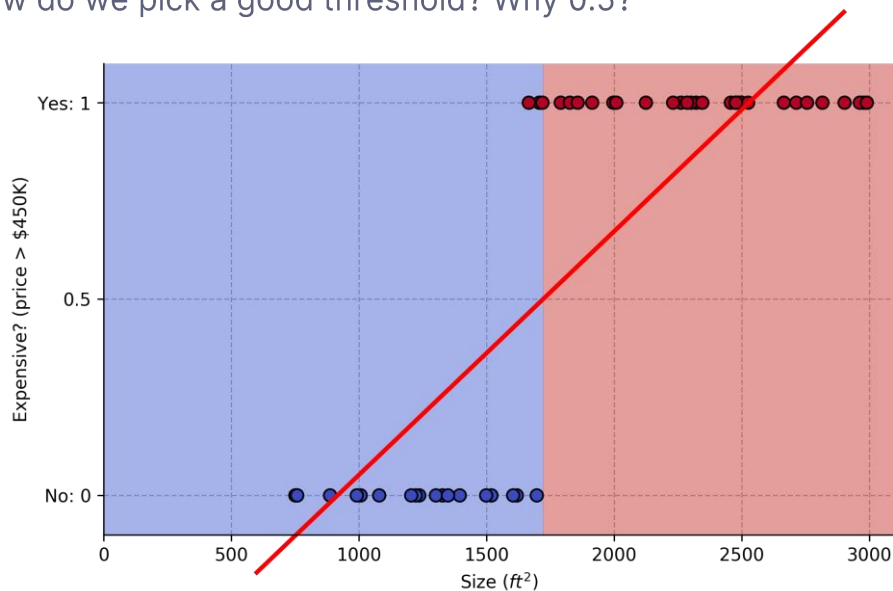
Regression vs. Classification

- Can we use Linear Regression to do Classification?



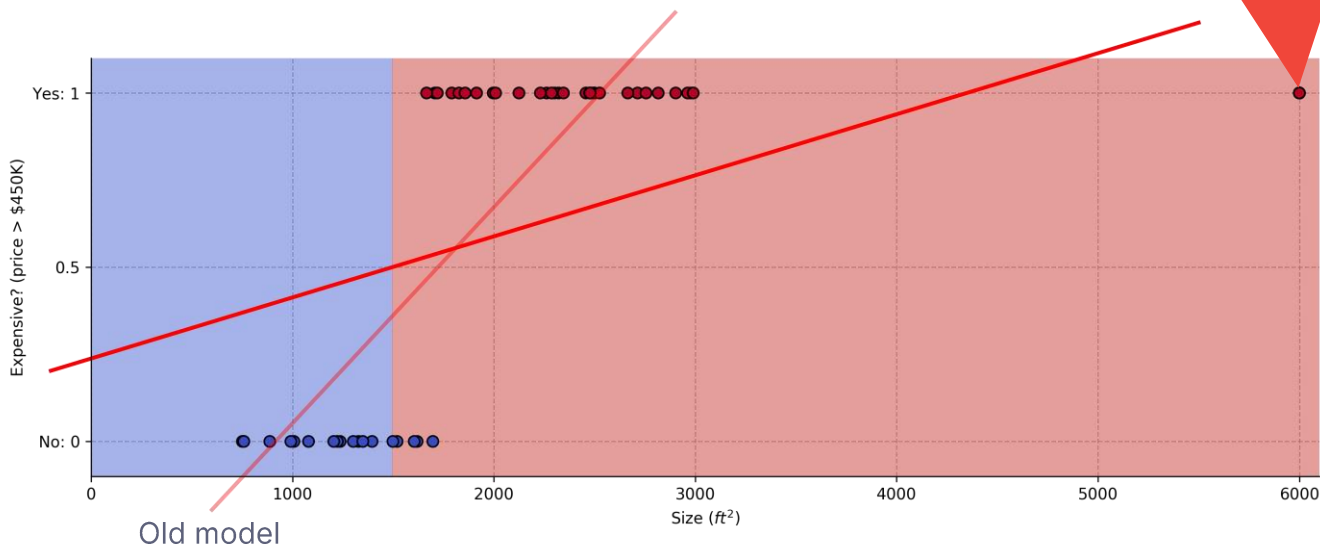
Regression vs. Classification

- Can we use Linear Regression to do Classification?
 - Not too bad, but a little hard to interpret the results.
 - Also, how do we pick a good threshold? Why 0.5?



Regression vs. Classification

- Same dataset as before, but with an added point.
 - Keeping the same 0.5 threshold makes some points get misclassified.
 - Can we find a better solution?



Regression vs. Classification

- Same dataset as before, but with an added point.
 - Keeping the same 0.5 threshold makes some points get misclassified.
 - Can we find a better solution? → **Logistic Regression**

