



# EN3150 Pattern Recognition

## Linear Models for Regression

---

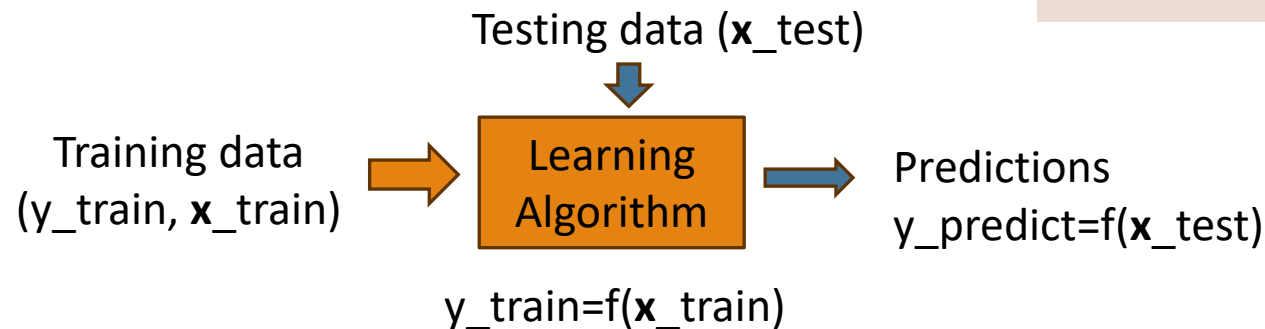
M. T. U. Sampath K. Perera,  
Department of Electronic and Telecommunication Engineering,  
University of Moratuwa.  
(sampathk@uom.lk).  
Semester 5 – Batch 20.

# What is regression ?

Predict the value of dependent variable based on the values of one or more independent variables

- Regression is a supervised learning technique

Dependent variable is often a quantitative or continuous variable



- Applications
  - Trend forecasting (e.g., forecast future sales)
  - Forecasting an effect (e.g., predicting the effect of advertising spending on sales)
  - To determine the strength of predictors (regression coefficients can indicate the magnitude and direction of influence each predictor )

# Key questions

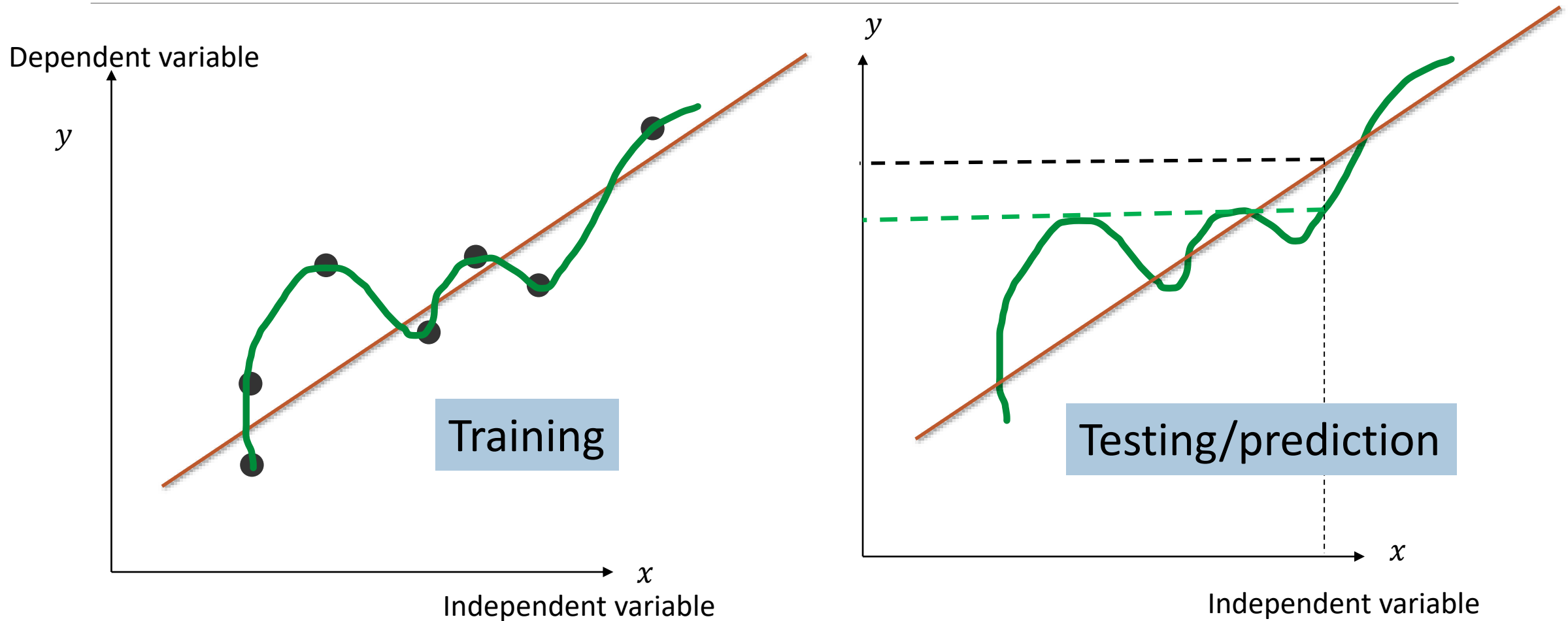
---

- Is there a relationship between dependent variable and independent variable(s)?
- How strong is the relationship ? and is it positive or negative relationship ? Linear or non-linear?
- How each feature/ independent variable contributes to the dependent variable, which are strong?
- How accurate our predictions?
- Is there synergy among features/ independent variables

Diabetes dataset: independent variable(s) <a href="#">Diabetes Data (ncsu.edu)</a>	
age	tc total serum cholesterol
sex	ldl, low-density lipoproteins
body mass index	hdl, high-density lipoproteins
average blood pressure	ltg, possibly log of serum triglycerides level
tch, total cholesterol / HDL	glu, blood sugar level

Dependent variable:  
quantitative measure of  
disease progression one  
year after baseline

# What is regression ?



# Linear Regression

---

- Most basic form of regression
- Relationship between the dependent and independent variables is assumed to be linear.

$$y(\mathbf{x}) \approx w_0 + w_1 x_1 + \dots + w_D x_D$$
$$y(\mathbf{x}) = w_0 + w_1 x_1 + \dots + w_D x_D + \epsilon = \mathbf{w}^T \mathbf{x} + \epsilon$$

$$\mathbf{x} = \begin{bmatrix} 1 \\ x_1 \\ \cdot \\ \cdot \\ \cdot \\ x_D \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \cdot \\ \cdot \\ \cdot \\ w_D \end{bmatrix}$$

- $w_0$  - Intercept or bias term
- What bias represents?
- $\epsilon$  – Error term, Why this term?
- $\mathbf{w}_{(1:D)}$  – Weights or regression coefficients (**parameters**) of independent variables

# Linear Regression

$$y(\mathbf{x}) = w_0 + w_1 x_1 + \dots + w_D x_D + \epsilon = \mathbf{w}^T \mathbf{x} + \epsilon$$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_N \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_D \end{bmatrix} \quad \mathbf{x}_i = \begin{bmatrix} 1 \\ x_{1,i} \\ \vdots \\ x_{D,i} \end{bmatrix}$$

➤  $\mathbf{w}^T \mathbf{x}$  = inner or scalar product between the input vector  $\mathbf{x}$  and the weight vector  $\mathbf{w}$

➤  $y(\mathbf{x})$  is dependent variable (continuous)

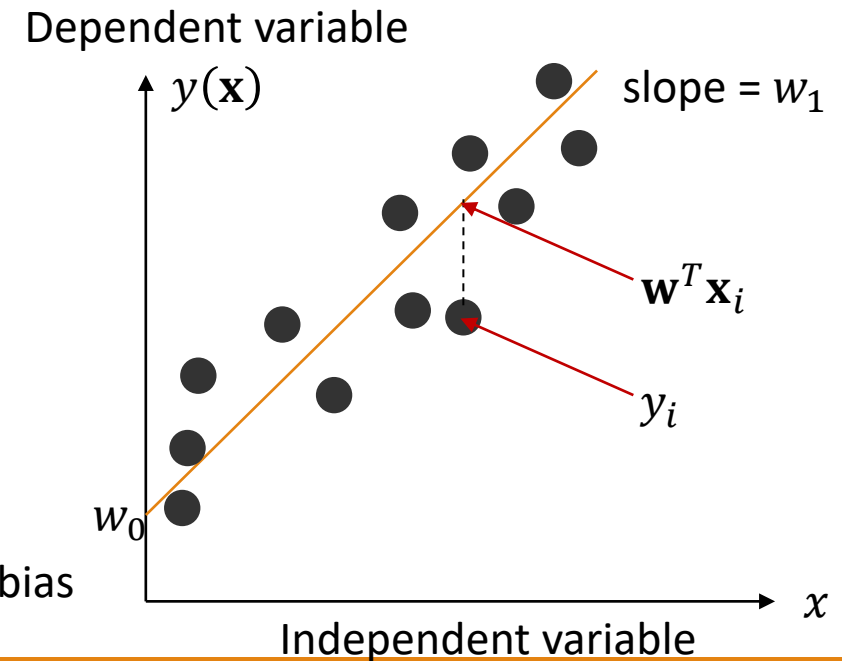
➤  $\mathbf{w}$  To be learned from data. How?

➤ Cost function

$$J(\mathbf{w}) = \sum_{i=1}^N (y_i - \mathbf{w}^T \mathbf{x}_i)^2$$

➤ Objective: Find  $\mathbf{w}$  which has the minimum  $J(\mathbf{w})$  for our training samples (N number of samples)

➤ How?



# Linear Regression

---

➤  $y(\mathbf{x}) = w_0 + w_1 x_1 + \dots, + w_D x_D + \epsilon = \mathbf{w}^T \mathbf{x} + \epsilon$

➤ Hypothesis:  $y(\mathbf{x}) = w_0 + w_1 x_1 + \dots, + w_D x_D$

➤ Parameters :  $w_0, w_1, \dots, w_D$

➤ Cost function :  $J(\mathbf{w}) = \sum_{i=1}^N (y_i - \mathbf{w}^T \mathbf{x}_i)^2$

➤ Objective: Find  $\mathbf{w}$  which has the minimum  $J(\mathbf{w})$

$$\hat{\mathbf{w}} = \arg \min_{\mathbf{w}} J(\mathbf{w})$$

# Linear Regression

## ➤ Cost function

$$J(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^N (y_i - \mathbf{w}^T \mathbf{x}_i)^2$$

➤ Objective: Find which has the minimum  $J(\mathbf{w})$  for our training samples (N number of samples)

➤  $\hat{\mathbf{w}} = \arg \min_{\mathbf{w}} J(\mathbf{w})$

Gradient descent

Update for  $w_j$   $w_j \leftarrow w_j - \alpha \frac{\partial J(\mathbf{w})}{\partial w_j}$

learning rate

For one training sample

$$\frac{\partial J(\mathbf{w})}{\partial w_j} = \frac{1}{2} \frac{\partial (y_i - \mathbf{w}^T \mathbf{x}_i)^2}{\partial w_j} = \frac{2(y_i - \mathbf{w}^T \mathbf{x}_i)}{2} \frac{\partial (y_i - \mathbf{w}^T \mathbf{x}_i)}{\partial w_j} = (y_i - \mathbf{w}^T \mathbf{x}_i)(-\mathbf{x}_i)$$

Update rule

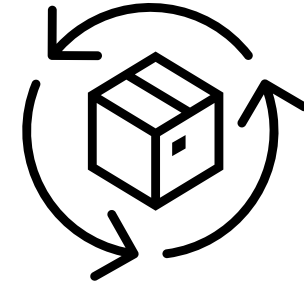
$$w_j \leftarrow w_j + \alpha \cdot \sum_{i=1}^N (y_i - \mathbf{w}^T \mathbf{x}_i) \mathbf{x}_i$$

If model output and true value are close, we need little update on parameter

Batch gradient descent







# Linear Regression

---

Initialize  $\mathbf{w}$

Repeat until convergence {

$$w_j \leftarrow w_j + \alpha \sum_{i=1}^N (y_i - \mathbf{w}^T \mathbf{x}_i) \mathbf{x}_i$$

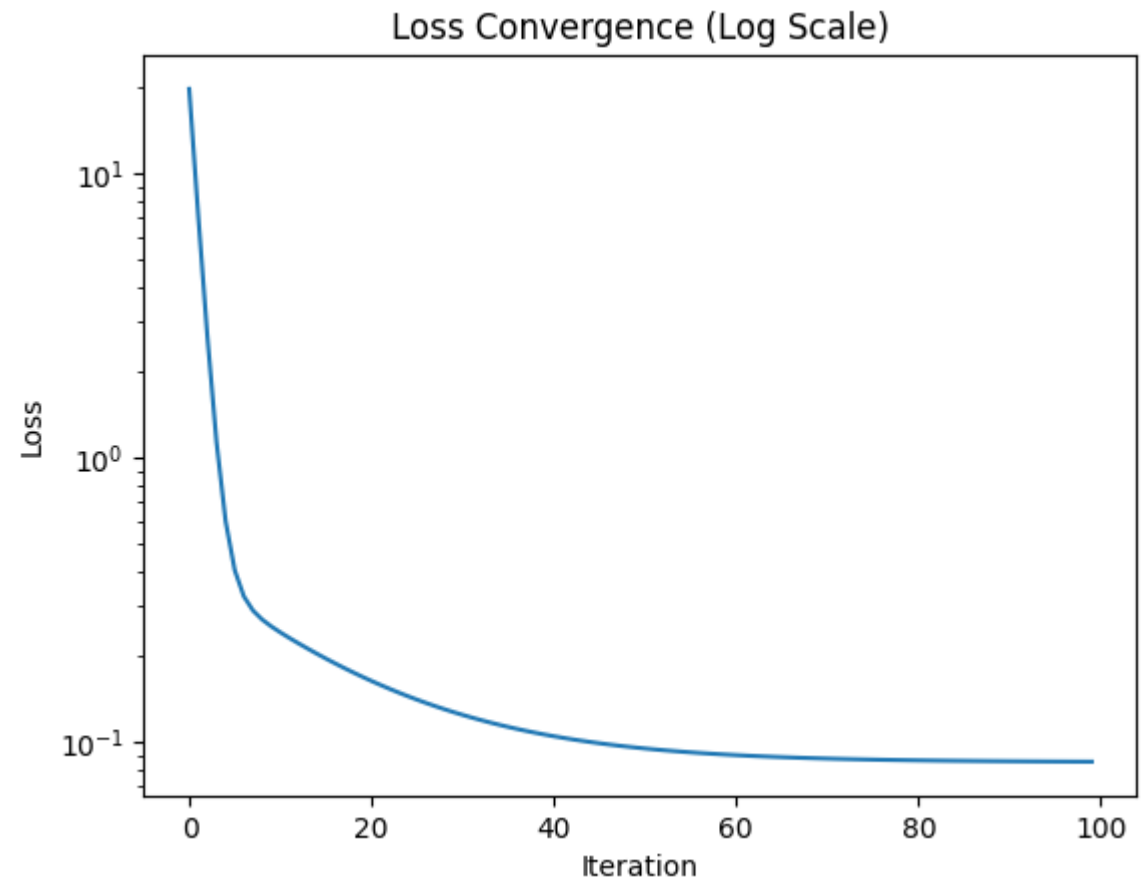
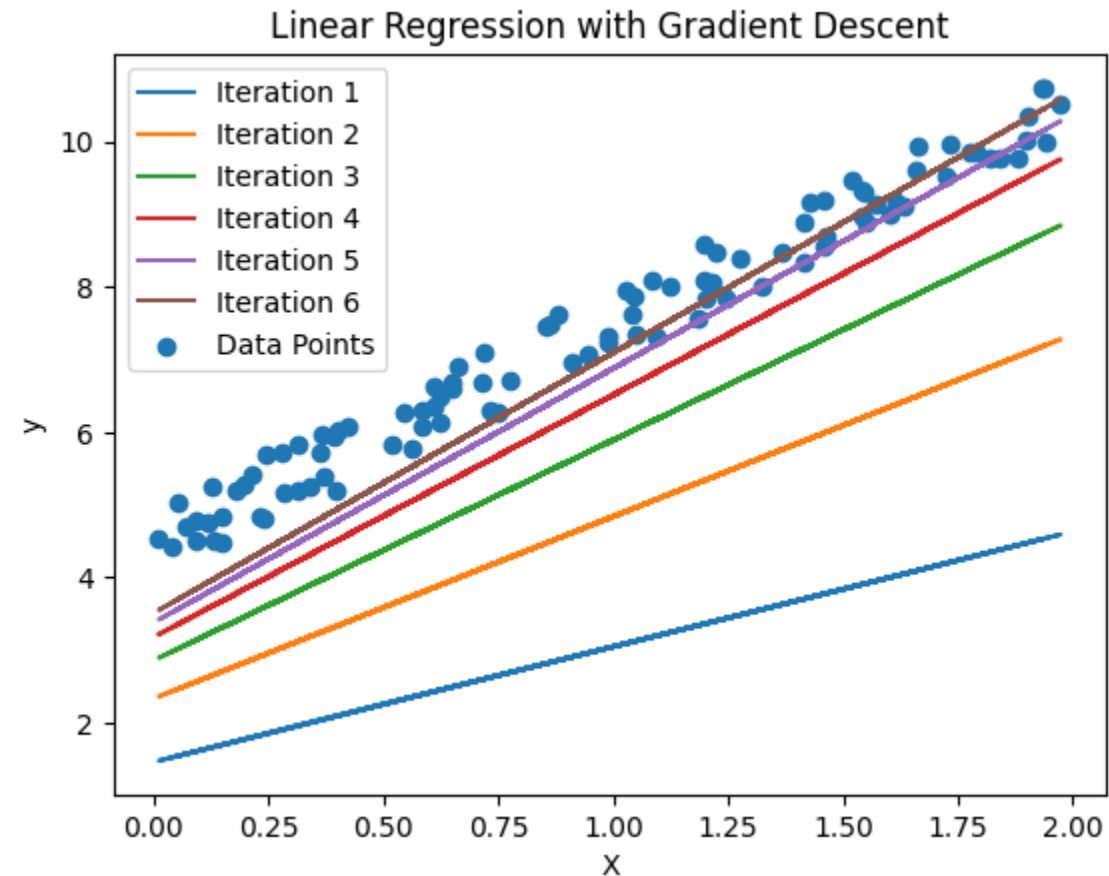
for every  $j$

}

iterative algorithm

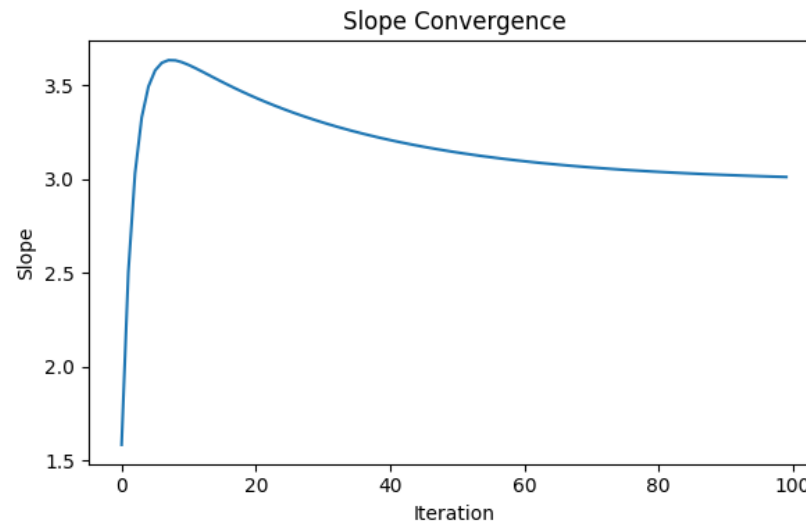
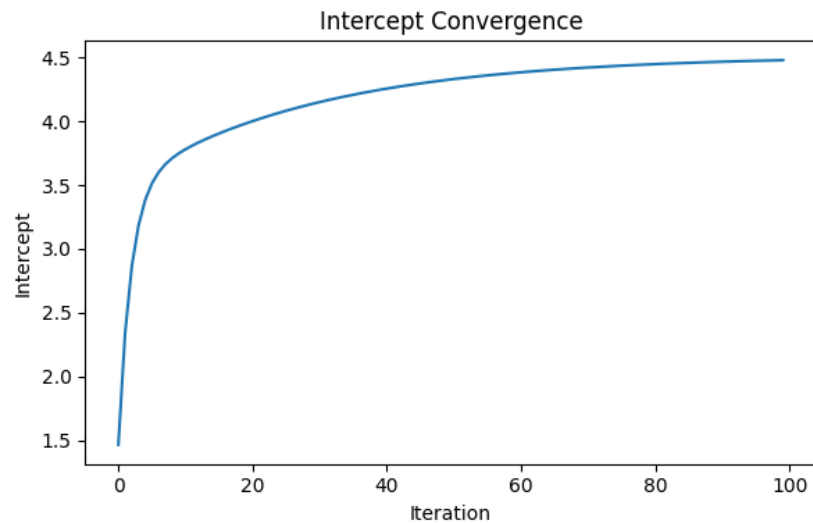
# Linear Regression

$y = 4 + 3x$ , Initialization  $w_0, w_1 = [0, 0]$ , learning rate  $= 0.001$

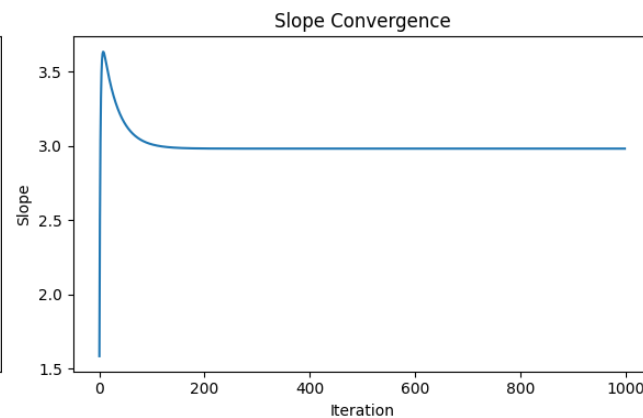
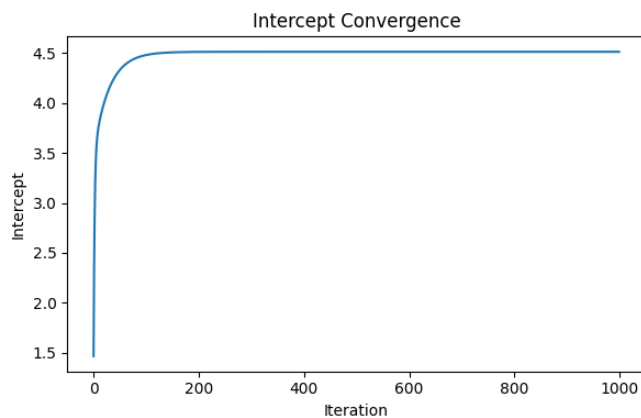


# Linear Regression

$y = 4 + 3x$ , Initialization  $w_0, w_1 = [0, 0]$ , learning rate  $= 0.001$



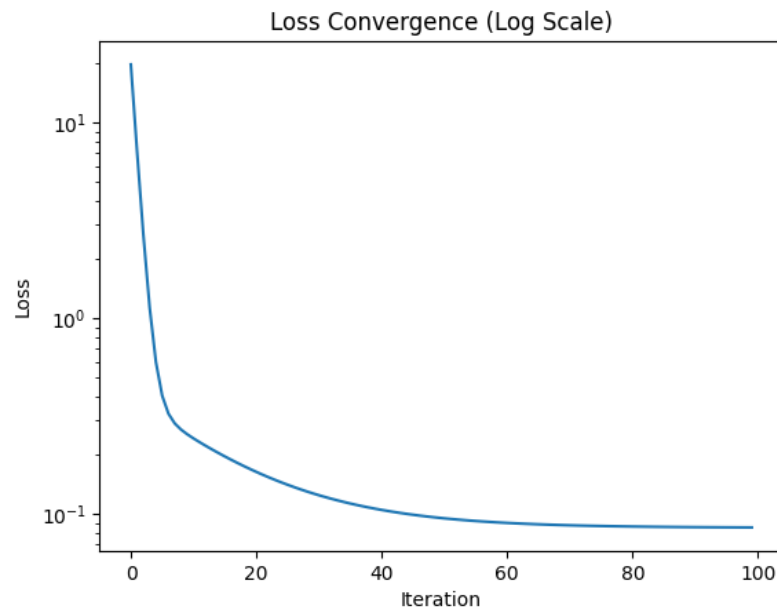
After 100 iterations  
Intercept: 4.48  
Slope: 3.01



# Linear Regression

$y = 4 + 3x$ , learning rate  $= 0.001$

Initialization  $w_0, w_1 = [0, 0]$ ,

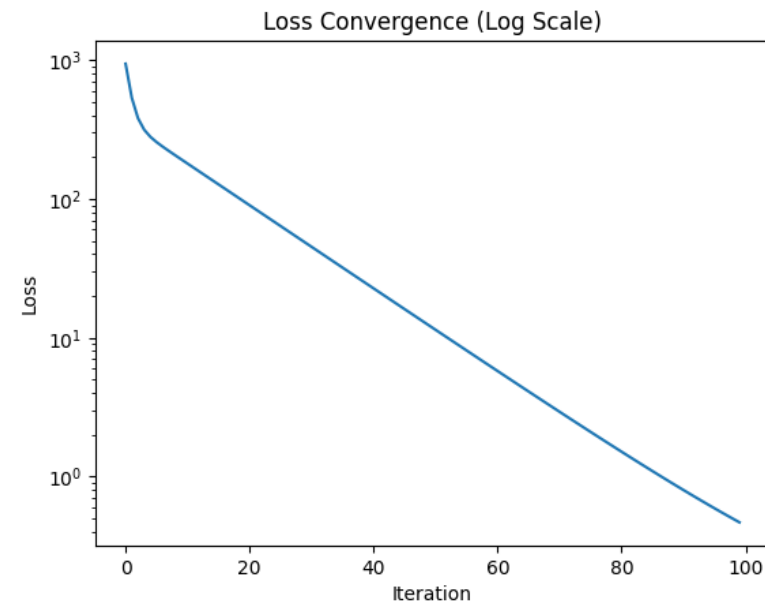


After 100 iterations

Intercept: 4.48

Slope: 3.01

Initialization  $w_0, w_1 = [-50, 13]$ ,



After 100 iterations

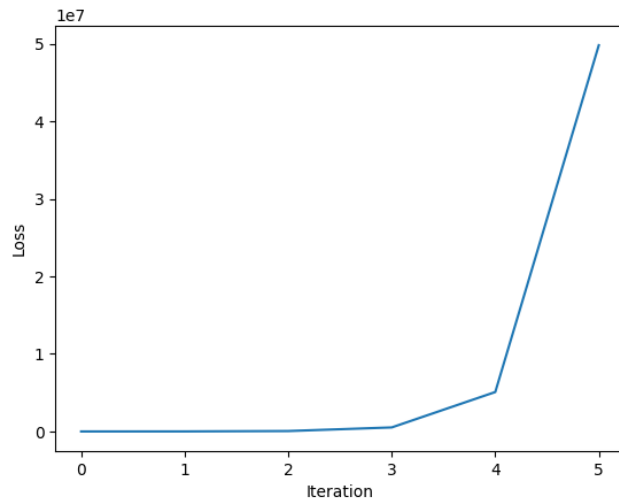
Intercept: 3.38

Slope: 3.97

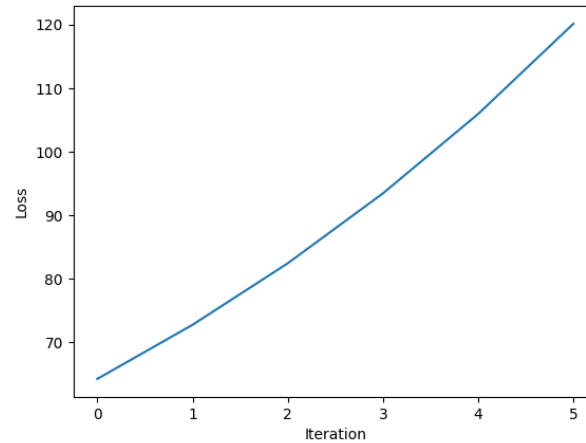
# Linear Regression

$Y = 4 + 3x$ , Initialization  $w_0, w_1 = [0, 0]$ ,

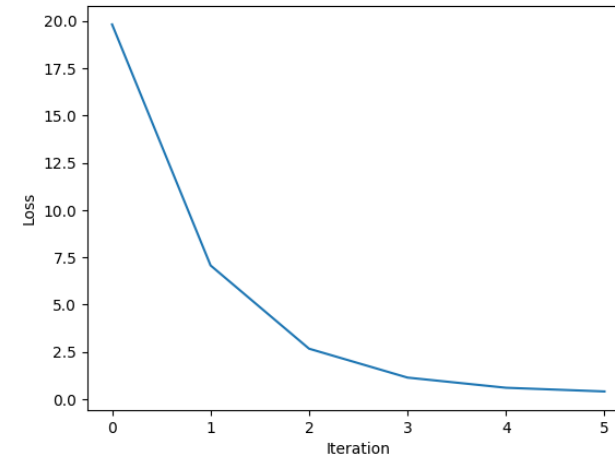
learning rate = 0.01



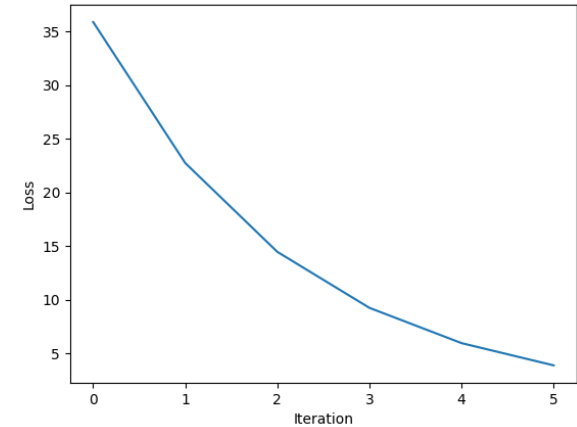
learning rate = 0.005



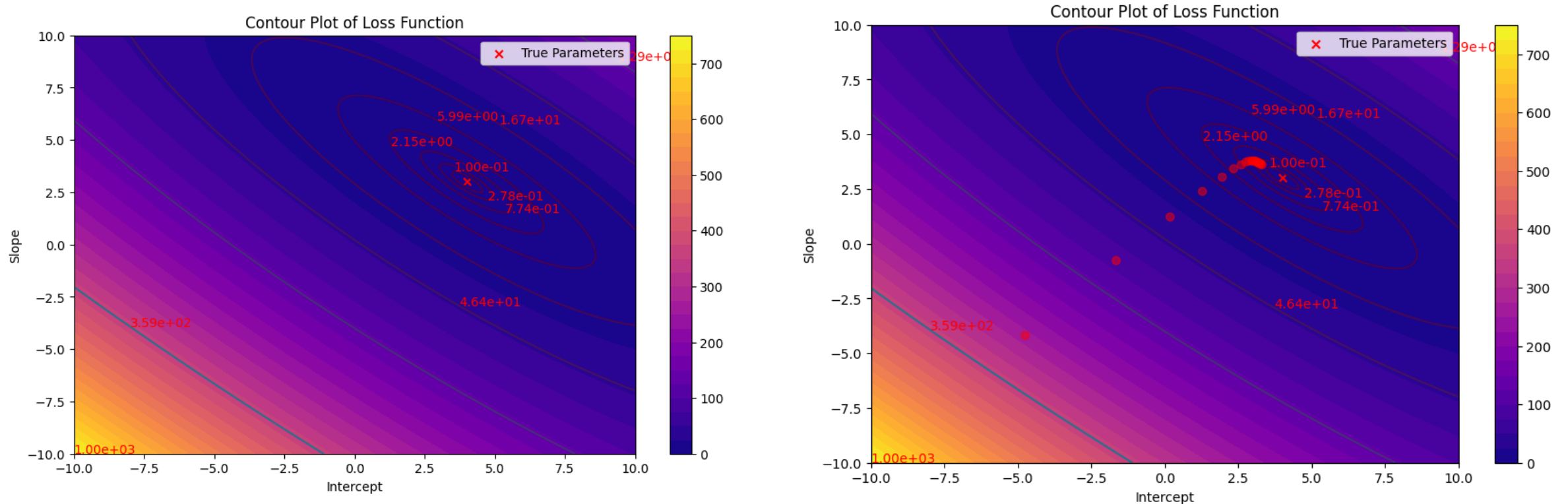
learning rate = 0.001



learning rate = 0.0005



# Linear Regression



A contour plot is a graphical representation used to visualize a function  $F(x,y)$  of two variables. In this type of plot, distinct colors are assigned to different values of  $F$ . You'll observe a set of curves on the graph. These curves are called contours and are drawn in such a way that they follow paths along which the values of  $F(x,y)$  remain constant. Each contour line corresponds to a specific value of  $F$ .

# Gradient descent variants

---

## Batch Gradient Descent

- Calculate the gradient (derivative) of the objective function with respect to all the training data points
- Slow but more accurate

## Stochastic Gradient Descent (SGD)

- SGD uses only one randomly selected data point at a time
- Faster convergence
- Can be noisy and might converge to a suboptimal solution

## Mini-batch Gradient Descent:

- Computes the gradient using a small randomly selected subset (mini-batch) of the dataset

# Linear Regression:

- For one independent variable (one dimensional,  $D=1$ )

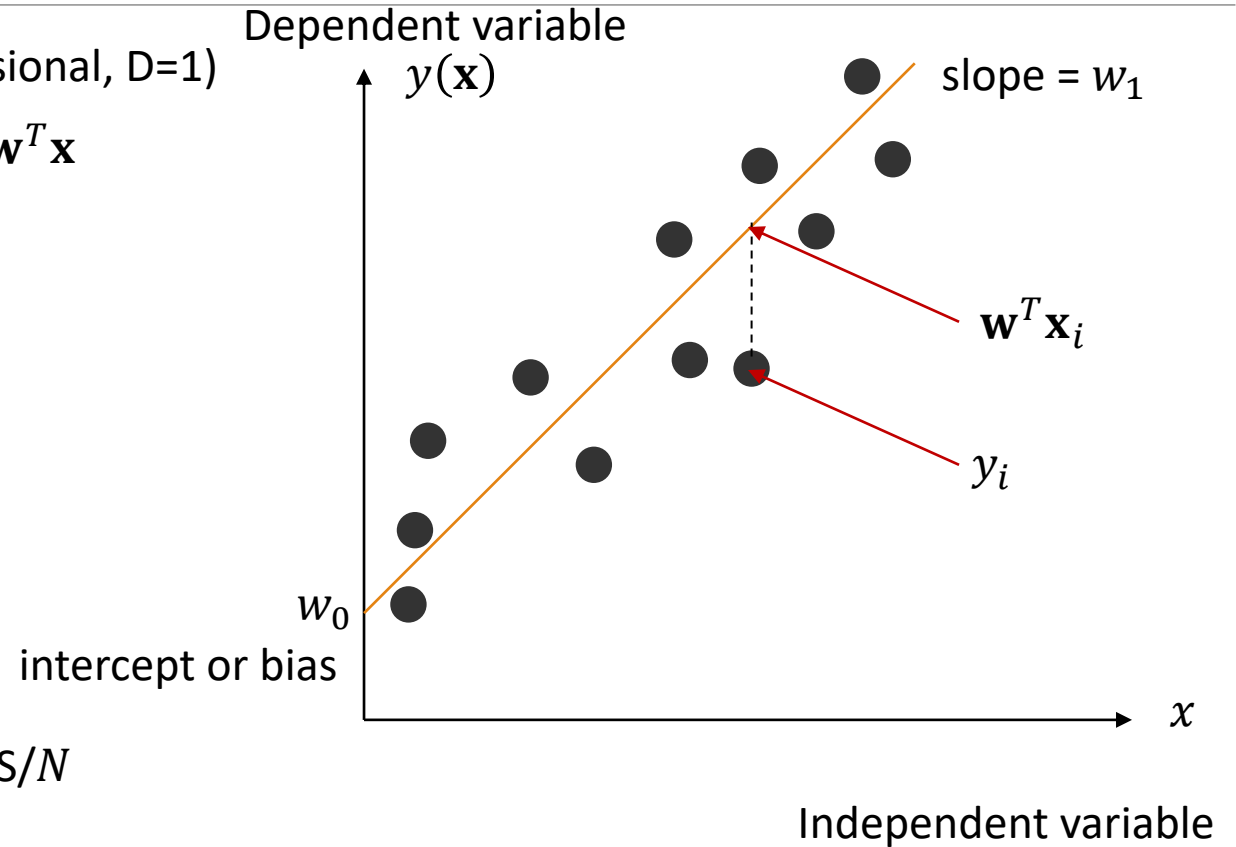
$$y(\mathbf{x}) = w_0 + w_1 x_1 + \dots + w_D x_D = \mathbf{w}^T \mathbf{x}$$

$$y(\mathbf{x}) = w_0 + w_1 x_1$$

Residual Sum of Squares (RSS) or  
Sum of Squared Error (SSE)

$$\text{RSS}(\mathbf{w}) \triangleq \sum_{i=1}^N (y_i - \mathbf{w}^T \mathbf{x}_i)^2$$

Mean Squared Error (MSE) =  $\text{SSE}/N$  or  $\text{RSS}/N$





# Linear Regression: Normal equations

$$\begin{aligned}
 \text{RSS}(\mathbf{w}) &\triangleq \sum_{i=1}^N (y_i - \mathbf{w}^T \mathbf{x}_i)^2 = \mathbf{y} - [w_0 \ w_1 \ \dots \ w_D] \begin{bmatrix} 1 \\ x_{1,1} \\ \vdots \\ x_{D,1} \end{bmatrix} - [w_0 \ w_1 \ \dots \ w_D] \begin{bmatrix} 1 \\ x_{1,2} \\ \vdots \\ x_{D-1,2} \end{bmatrix} - \dots - [w_0 \ w_1 \ \dots \ w_D] \begin{bmatrix} 1 \\ x_{1,N} \\ \vdots \\ x_{D,N} \end{bmatrix} \\
 &= \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 \\
 &= (\mathbf{y} - \mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w})
 \end{aligned}$$

$$\mathbf{X} = \begin{bmatrix} 1 & x_{1,1} & x_{D,1} \\ 1 & x_{1,2} & x_{D,2} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ 1 & x_{1,N} & x_{D,N} \end{bmatrix}$$

➤  $\mathbf{w}$  are estimated by minimizing the MSE.

➤ How ? Setting the gradient to zero

➤ Gradient  $\mathbf{g}(\mathbf{w}) = (\mathbf{X}^T \mathbf{X})\mathbf{w} - \mathbf{X}^T \mathbf{y}$

➤  $\mathbf{g}(\mathbf{w}) = 0 \Rightarrow (\mathbf{X}^T \mathbf{X})\mathbf{w} = \mathbf{X}^T \mathbf{y}$  normal equations

➤  $\hat{\mathbf{w}}_{\text{OLS}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$  Known as ordinary least squares

# Linear Regression

Example:

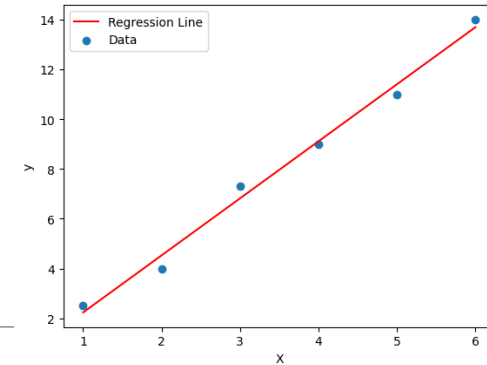
X (independent variable)	y
1	2.5
2	4
3	7.3
4	9
5	11
6	14

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} 1 & x_{1,1} & x_{D,1} \\ 1 & x_{1,2} & x_{D,2} \\ \vdots & \vdots & \vdots \\ 1 & x_{1,N} & x_{D,N} \end{bmatrix}$$

$$\hat{\mathbf{w}}_{\text{OLS}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

1. Write down  $\mathbf{X}$  and  $\mathbf{y}$
2. What is  $\hat{\mathbf{w}}_{\text{OLS}}$ ?
3. Predict value of  $y$  for  $x=8$ .

# Linear Regression



x	y
1	2.5
2	4
3	7.3
4	9
5	11
6	14

$$\mathbf{X} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \\ 1 & 6 \end{bmatrix}$$

$$\mathbf{y} = [2.5 \ 4 \ 7.3 \ 9 \ 11 \ 14]$$

$$\mathbf{X}^T \mathbf{y} = \begin{bmatrix} 47.8 \\ 207.4 \end{bmatrix}$$

$$(\mathbf{X}^T \mathbf{X}) = \begin{bmatrix} 6 & 21 \\ 21 & 91 \end{bmatrix}$$

$$(\mathbf{X}^T \mathbf{X})^{-1} = \begin{bmatrix} 0.86 & -0.2 \\ -0.2 & 0.057 \end{bmatrix}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

1. Write down  $\mathbf{X}$  and  $\mathbf{y}$
2. What is  $\hat{\mathbf{w}}_{\text{OLS}}$ ?
3. Predict value of  $y$  for  $x=8$ .

$$\begin{aligned} \hat{\mathbf{w}}_{\text{OLS}} &= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \\ &= \begin{bmatrix} 0.86 & -0.2 \\ -0.2 & 0.057 \end{bmatrix} \begin{bmatrix} 47.8 \\ 207.4 \end{bmatrix} = \begin{bmatrix} -0.053 \\ 2.29 \end{bmatrix} \end{aligned}$$


$$\hat{y} = -0.053 + 2.29 \times 8 = 18.267$$

# Linear Regression

---

- $y(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + \epsilon \rightarrow \epsilon = y(\mathbf{x}) - \mathbf{w}^T \mathbf{x}$
- Gaussian noise model: assume that error term  $\epsilon$  follows a Gaussian distribution (zero mean and variance  $\sigma^2$ )  $p(\epsilon) = \mathcal{N}(0 | \sigma^2)$
- $p(y|\mathbf{x}, \mathbf{w}) = \mathcal{N}(y | \mathbf{w}^T \mathbf{x}, \sigma^2)$

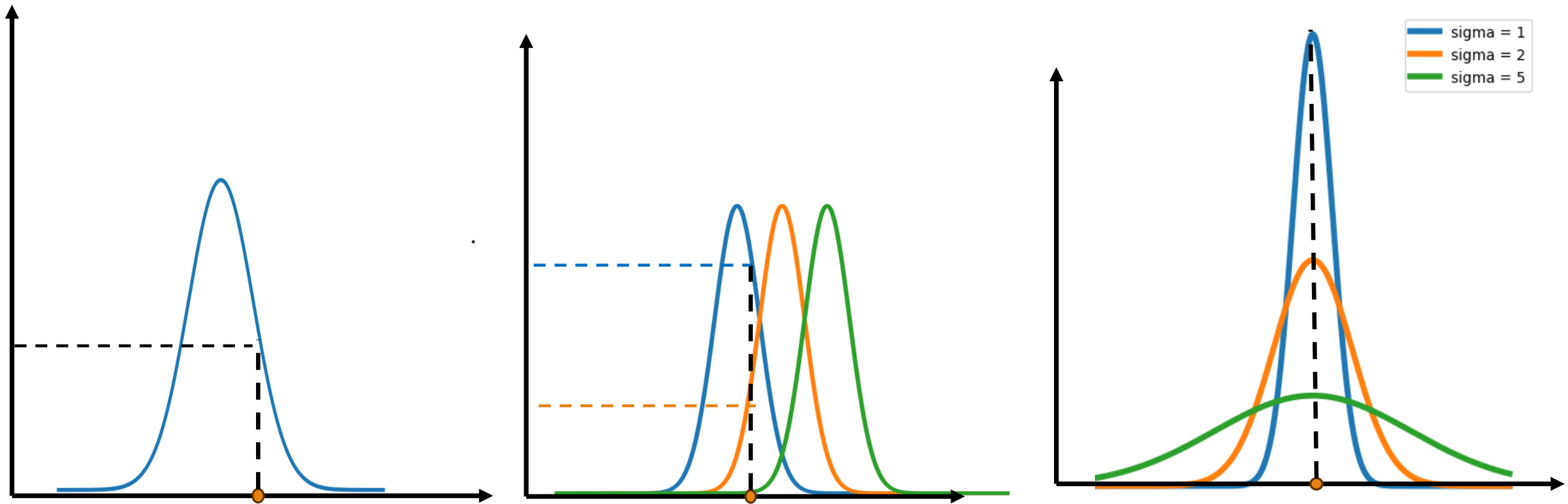
Mean  $\mathbf{w}^T \mathbf{x}$  and variance  $\sigma^2$



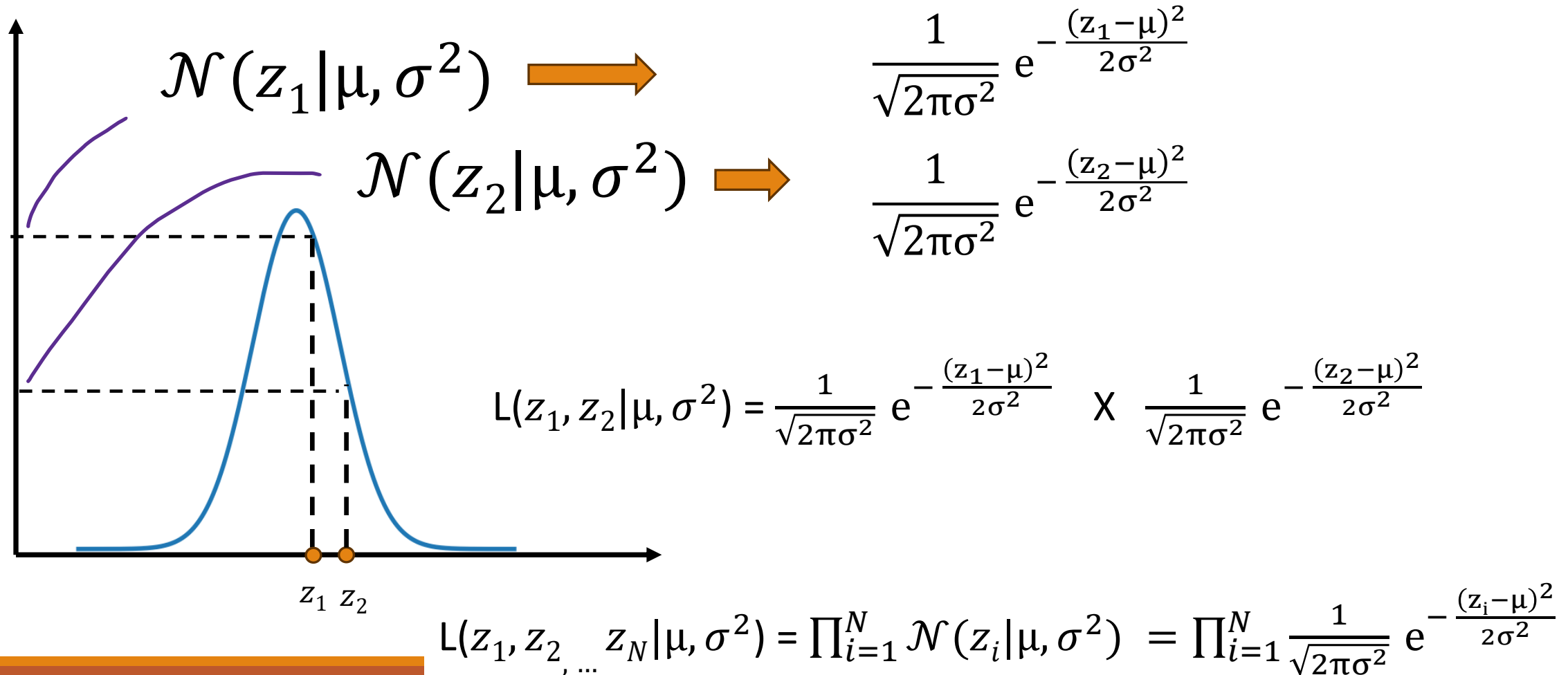
conditional probability of the output variable  $y$  given the input variable  $\mathbf{x}$  and model parameters  $\mathbf{w}$

# Linear Regression: Likelihood

---



# Linear Regression: Likelihood



# Linear Regression: Likelihood

---

$$L(y_1, y_2, \dots, y_N | \mathbf{w}^T \mathbf{x}, \sigma^2) = \prod_{i=1}^N \mathcal{N}(y_i | \mathbf{w}^T \mathbf{x}_i, \sigma^2)$$

$$\text{Log likelihood} = \ln \left( \prod_{i=1}^N \mathcal{N}(y_i | \mathbf{w}^T \mathbf{x}_i, \sigma^2) \right) = \ln \left( \prod_{i=1}^N \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i - \mu)^2}{2\sigma^2}} \right)$$

$$= N \ln \left( \frac{1}{\sqrt{2\pi\sigma^2}} \right) - \sum_{i=1}^N \frac{(y_i - \mu)^2}{2\sigma^2}$$

$$= -\frac{N}{2} \ln(\sqrt{2\pi}) - \frac{N}{2} \ln(\sigma) - \sum_{i=1}^N \frac{(y_i - \mu)^2}{2\sigma^2}$$

# Linear Regression: Likelihood

$$L(y_1, y_2, \dots, y_N | \mathbf{w}^T \mathbf{x}, \sigma^2) = -\frac{N}{2} \ln(\sqrt{2\pi}) - \frac{N}{2} \ln(\sigma) - \sum_{i=1}^N \frac{(y_i - \mu)^2}{2\sigma^2}$$

$\mu = (\mathbf{w}^T \mathbf{x})$  and  $\mathbf{x}$  is given too.

In vector-matrix form (with a  $\frac{1}{2}$ , NLL – Negative Log Likelihood)

$$\text{NLL}(\mathbf{w}) = \frac{1}{2} (\mathbf{y} - \mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w})$$

- In this case, NLL ( $\mathbf{w}$ ) is equivalent to **residual sum of squares (RSS)\***
- How to find  $\mathbf{w}$  (assuming  $\sigma$  is fixed), Setting the gradient to zero
- Gradient  $\mathbf{g}(\mathbf{w}) = (\mathbf{X}^T \mathbf{X})\mathbf{w} - \mathbf{X}^T \mathbf{y}$
- $\mathbf{g}(\mathbf{w}) = 0 \Rightarrow (\mathbf{X}^T \mathbf{X})\mathbf{w} = \mathbf{X}^T \mathbf{y}$
- $\hat{\mathbf{w}}_{\text{OLS}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$

Maximum likelihood estimator (MLE)\*\*

\*with certain constant factors in the expression are disregarded

\*\*Minimizing the negative log-likelihood is equivalent to maximizing the likelihood



# Linear Regression

---

$$\hat{\mathbf{w}}_{\text{OLS}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} = \mathbf{X}^\dagger \mathbf{y}$$

pseudo inverse of the (non-square) matrix  $\mathbf{X}$

➤ Condition for unique solution

➤  $\mathbf{X}$  is a full rank matrix\* (columns of  $\mathbf{X}$  are linearly independent)

➤ Full rank matrix ➔ No row or column can be expressed as a linear combination of the other rows or columns, respectively

$$\mathbf{X} = \begin{bmatrix} 1 & x_{1,1} & x_{D,1} \\ 1 & x_{1,2} & x_{D,2} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ 1 & x_{1,N} & x_{D,N} \end{bmatrix}$$

# Linear Regression: Likelihood

---

➤ Deriving the MLE for  $\sigma^2$

$$L(y_1, y_2, \dots, y_N | \hat{\mathbf{w}}^T \mathbf{x}, \sigma^2) = -\frac{N}{2} \ln(\sqrt{2\pi}) - \frac{N}{2} \ln(\sigma) - \sum_{i=1}^N \frac{(y_i - \hat{\mathbf{w}}^T \mathbf{x}_i)^2}{2\sigma^2}$$

$$\sigma^2 = \arg \min_{\sigma^2} \text{NLL}(\hat{\mathbf{w}}, \sigma^2) = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{\mathbf{w}}^T \mathbf{x}_i)^2$$

# Linear Regression: Solving $w_0$ separately

➤  $\mathcal{N}(y | w_0 + w_1 x_1 + \dots, + w_D x_D, \sigma^2)$

➤ 
$$L(y_1, y_2, \dots, y_N | \mathbf{w}^T \mathbf{x}, \sigma^2) = -\frac{N}{2} \ln(\sqrt{2\pi}) - \frac{N}{2} \ln(\sigma) - \sum_{i=1}^N \frac{(y_i - \mathbf{w}^T \mathbf{x}_i)^2}{2\sigma^2}$$

$$= -\frac{N}{2} \ln(\sqrt{2\pi}) - \frac{N}{2} \ln(\sigma) - \sum_{i=1}^N \frac{(y_i - (w_0 + w_1 x_{1i} + \dots, + w_D x_{Di}))^2}{2\sigma^2}$$

$$= -\frac{N}{2} \ln(\sqrt{2\pi}) - \frac{N}{2} \ln(\sigma) - \sum_{i=1}^N \frac{(y_i - (w_0 + \tilde{\mathbf{w}}^T \tilde{\mathbf{x}}_i))^2}{2\sigma^2}$$

➤ Solving  $w_0$  (Intercept or bias) and feature coefficients are separately

$$\frac{\partial L}{\partial w_0} = 0 \quad w_0 = \frac{1}{N} \sum_{i=1}^N (y_i - \tilde{\mathbf{w}}^T \tilde{\mathbf{x}}_i) = \bar{y} - \frac{1}{N} \sum_{i=1}^N (\tilde{\mathbf{w}}^T \tilde{\mathbf{x}}_i) = \bar{y} - \tilde{\mathbf{w}}^T \bar{\mathbf{x}}$$

Average of the  
model outcome

Average of the  
target values

# Linear Regression

➤  $\tilde{\mathbf{w}}^T = [w_1, \dots, w_D]$

➤ Solving feature coefficients are separately

➤  $\hat{\tilde{\mathbf{w}}} = \left[ \sum_{i=1}^N (\tilde{\mathbf{x}}_i - \bar{\mathbf{x}})(\tilde{\mathbf{x}}_i - \bar{\mathbf{x}})^T \right]^{-1} \left[ \sum_{i=1}^N (y_i - \bar{y})(\tilde{\mathbf{x}}_i - \bar{\mathbf{x}}) \right] = (\mathbf{X}_c^T \mathbf{X}_c)^{-1} \mathbf{X}_c^T \mathbf{y}_c$

Centered input matrix

Centered output vector

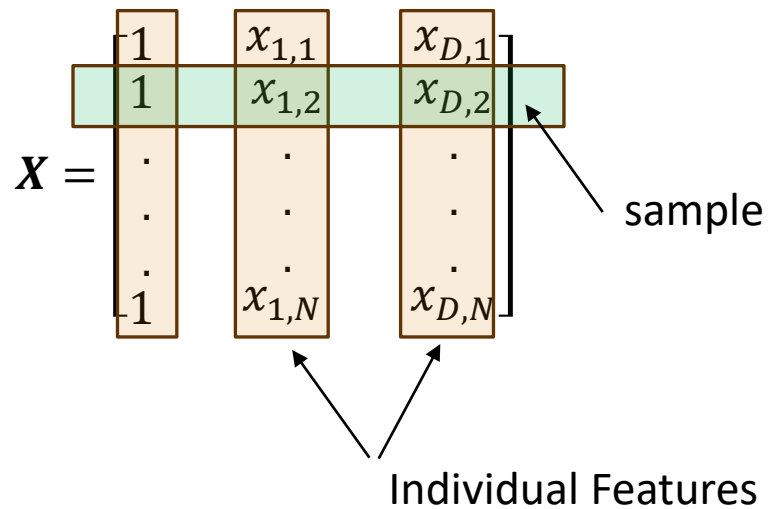
➤ For 1D  $y = w_0 + w_1 x$

$$\hat{w}_1 = \sum_{i=1}^N (y_i - \bar{y})(x_i - \bar{x}) / \sum_{i=1}^N (x_i - \bar{x})^2 = \text{Cov}[X; Y] / \text{Cov}[X; X] = C_{xy} / C_{xx}$$

$$\hat{w}_0 = \bar{y} - \hat{w}_1 \bar{x}$$

Procedure: Estimating  $\tilde{\mathbf{w}}$  by centered data then estimating the  $w_0$  by  $\bar{y} - \tilde{\mathbf{w}}^T \bar{\mathbf{x}}$

# Linear Regression: Geometric interpretation



$$\mathbf{X} = \begin{bmatrix} 1 & x_{1,1} & x_{D,1} \\ 1 & x_{1,2} & x_{D,2} \\ \vdots & \vdots & \vdots \\ 1 & x_{1,N} & x_{D,N} \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_D \end{bmatrix}$$

$$\mathbf{x}_i = \begin{bmatrix} 1 \\ x_{1,i} \\ \vdots \\ x_{D,i} \end{bmatrix}$$

$$\hat{\mathbf{w}}_{\text{OLS}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

$$\mathbf{Proj}(\mathbf{X}) = \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$$

$$\hat{\mathbf{y}} = \mathbf{Proj}(\mathbf{X}) \mathbf{y}$$

$$\hat{\mathbf{y}} = \mathbf{X} \hat{\mathbf{w}}_{\text{OLS}} = \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

The orthogonal projection of vector  $\mathbf{y}$  onto the column space of matrix  $\mathbf{X}$

# Linear Regression: Implementation

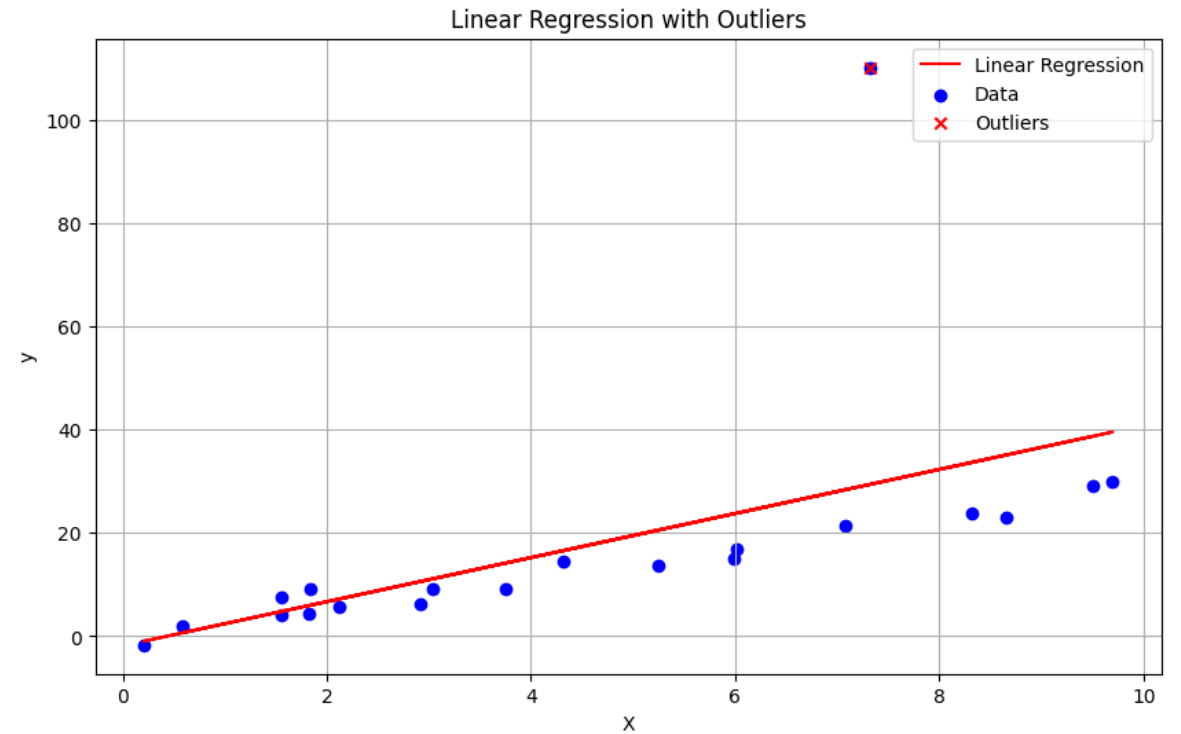
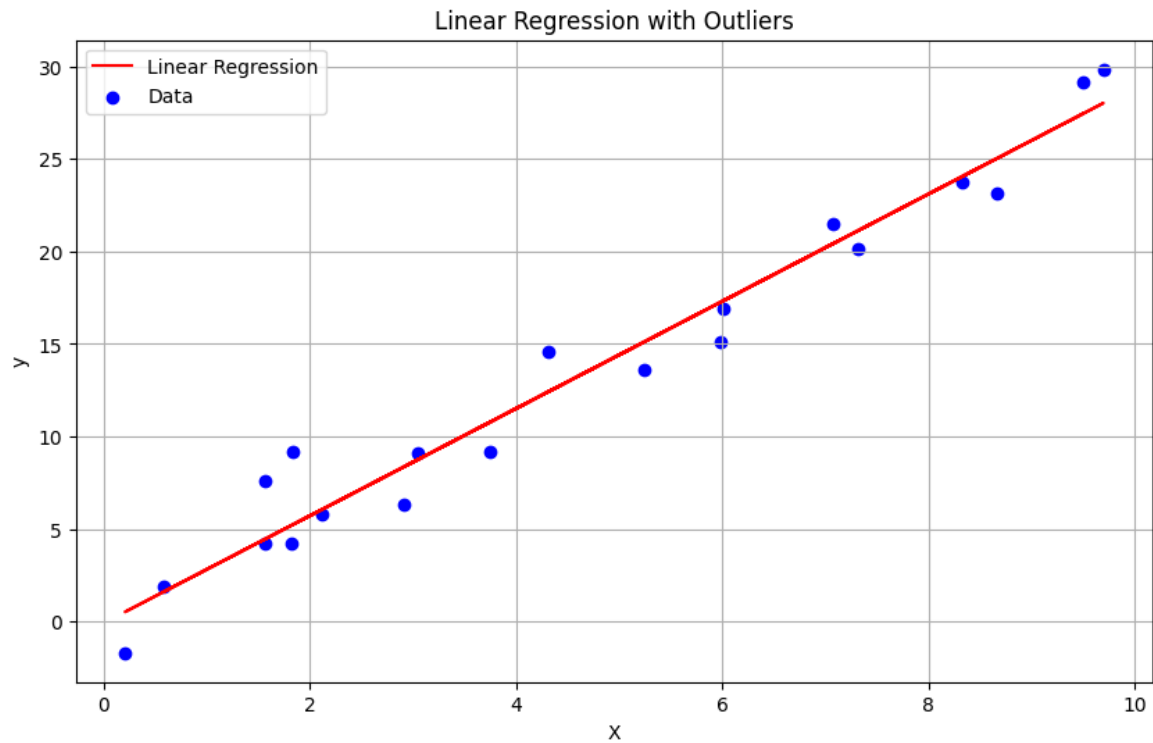
---

$$\hat{\mathbf{w}}_{\text{OLS}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

may be ill conditioned or singular → cause issues in numerical computations

- SVD based pseudo inverse calculation (see [sklearn implementation](#))

# Linear Regression: Example



# $R^2$ Statistic

---

“Amount of variability that is **left unexplained** after performing the regression (amount of unexplained)”

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_i - \mathbf{w}^T \mathbf{x})^2}{\sum_{i=1}^N (y_i - \bar{y})^2} = 1 - \frac{RSS}{TSS} = 1 - \frac{RSS}{TSS} \in (0, 1)$$

“Total variance in the  $y$ ”

TSS – RSS “measures the amount of variability in the response that is explained (or removed) by performing the regression” [1].

“ $R^2$  measures proportion of variability in  $Y$  that can be explained using  $X$ ” (“ $R^2$  is close to 1 mean that the large proportion of the variability in the response has been explained by the regression” [1].

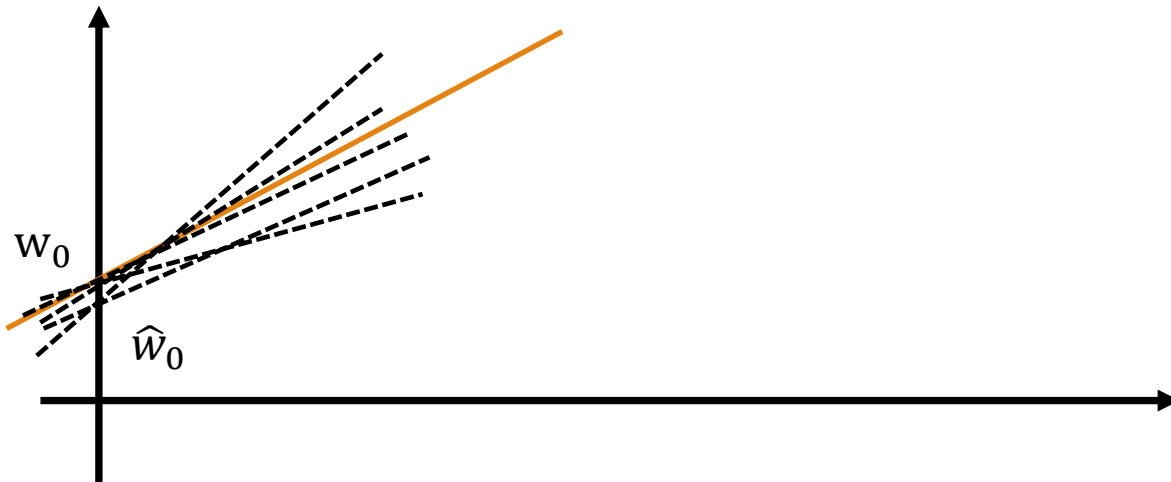


# Standard errors

- True relationship  $y = w_0 + w_1 x$
- How close our estimate  $(w_0, w_1)$  to true values  $(\hat{w}_0, \hat{w}_1)$
- Standard errors

$$SE^2(\hat{w}_0) = \sigma^2 \left[ \frac{1}{N} + \frac{\bar{x}^2}{\sum_{i=1}^N (x_i - \bar{x})^2} \right]$$

$$SE^2(\hat{w}_1) = \frac{\sigma^2}{\sum_{i=1}^N (x_i - \bar{x})^2}$$



# t-statistic

➤ Standard errors

$$SE^2(\hat{w}_0) = \sigma^2 \left[ \frac{1}{N} + \frac{\bar{x}^2}{\sum_{i=1}^N (x_i - \bar{x})^2} \right]$$

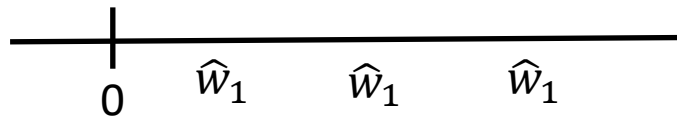
$$SE^2(\hat{w}_1) = \frac{\sigma^2}{\sum_{i=1}^N (x_i - \bar{x})^2}$$

➤ Is there a true relationship?

➤ Hypothesis testing

➤  $H_0$ : No relationship between  $y$  and  $w_1 \rightarrow w_1 = 0$

➤  $H_1$ : There is a relationship between  $y$  and  $w_1 \rightarrow w_1 \neq 0$



$$\text{t-statistic} \\ t = \frac{\hat{w}_1 - 0}{SE^2(\hat{w}_1)}$$

number of standard deviations  
that  $\hat{w}_1$  is away from 0

Need to determine  $\hat{w}_1$

1. Sufficiently far from zero  $\rightarrow \hat{w}_1 \neq 0$  or
2. Sufficiently *close to zero*  $\hat{w}_1 = 0$

$\hat{w}_1 = 0.1$  Sufficiently close to zero? Far away from zero?  
Depends on  $SE^2(\hat{w}_1)$

True  $w_1 = 5$ ,  $SE^2(\hat{w}_1) = 10$  is *significant*  
True  $w_1 = 1000$ ,  $SE^2(\hat{w}_1) = 10$  is not significant

# Confidence intervals

---

- Standard errors can be used to calculate confidence level.
- 95% Confidence Interval: Range of values containing true parameter value with 95% probability
- The range is calculated from training data.
- For linear regression

$$\text{For } w_0: \hat{w}_0 \pm 2 \text{ SE}^2(\hat{w}_0)$$

$$\text{For } w_1: \hat{w}_1 \pm 2 \text{ SE}^2(\hat{w}_1)$$

# $p$ -value

t-statistic

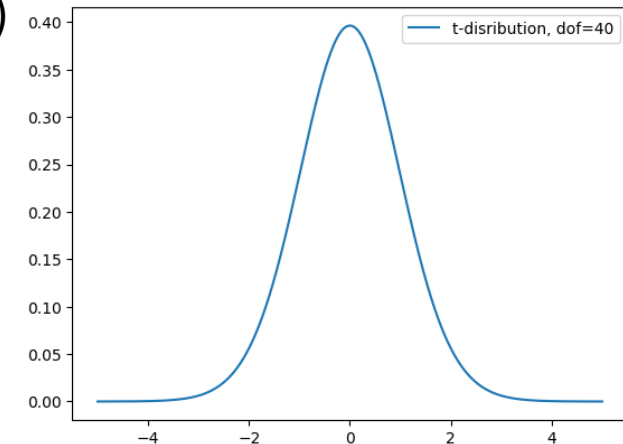
$$t = \frac{\hat{w}_1 - 0}{SE^2(\hat{w}_1)}$$

number of standard deviations  
that  $\hat{w}_1$  is away from 0

t-distribution with  $N - 2$  degrees of freedom if there  
is no relationship between  $x$  and  $y$

$p$ -value “the probability of observing any values equal to  $|t|$  or larger, assuming true  $w_1 = 0$ . (no association between the predictor and the response variable)”

small  $p$  -value reject null hypothesis  $H_0$ : No relationship between  $y$  and  $w_1$  ( $w_1 = 0$ )  
Typical values 1% or 5%.



# Linear Regression: Example

California housing dataset

#Attribute Information: (independent variables)

# - MedInc median income in block group

# - HouseAge median house age in block group

# - AveRooms average number of rooms per household

# - AveBedrms average number of bedrooms per household

# - Population block group population

# - AveOccup average number of household members

# - Latitude block group latitude

# - Longitude block group longitude

# Dependent variable is median house value (y)

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude
0	8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88	-122.23
1	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86	-122.22
2	7.2574	52.0	8.288136	1.073446	496.0	2.802260	37.85	-122.24
3	5.6431	52.0	5.817352	1.073059	558.0	2.547945	37.85	-122.25
4	3.8462	52.0	6.281853	1.081081	565.0	2.181467	37.85	-122.25

$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$
8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88	-122.23
8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86	-122.22
7.2574	52.0	8.288136	1.073446	496.0	2.802260	37.85	-122.24
5.6431	52.0	5.817352	1.073059	558.0	2.547945	37.85	-122.25
3.8462	52.0	6.281853	1.081081	565.0	2.181467	37.85	-122.25

$y$
4.526
3.585
3.521
3.413

$$X = \begin{bmatrix} 1 & x_{1,1} & x_{D,1} \\ 1 & x_{1,2} & x_{D,2} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ 1 & x_{1,N} & x_{D,N} \end{bmatrix}$$

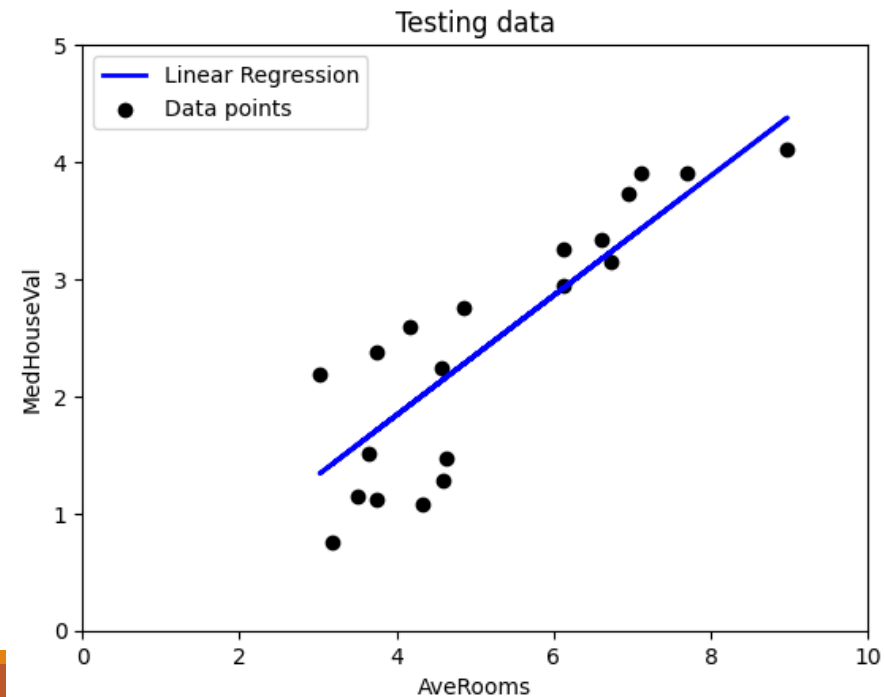
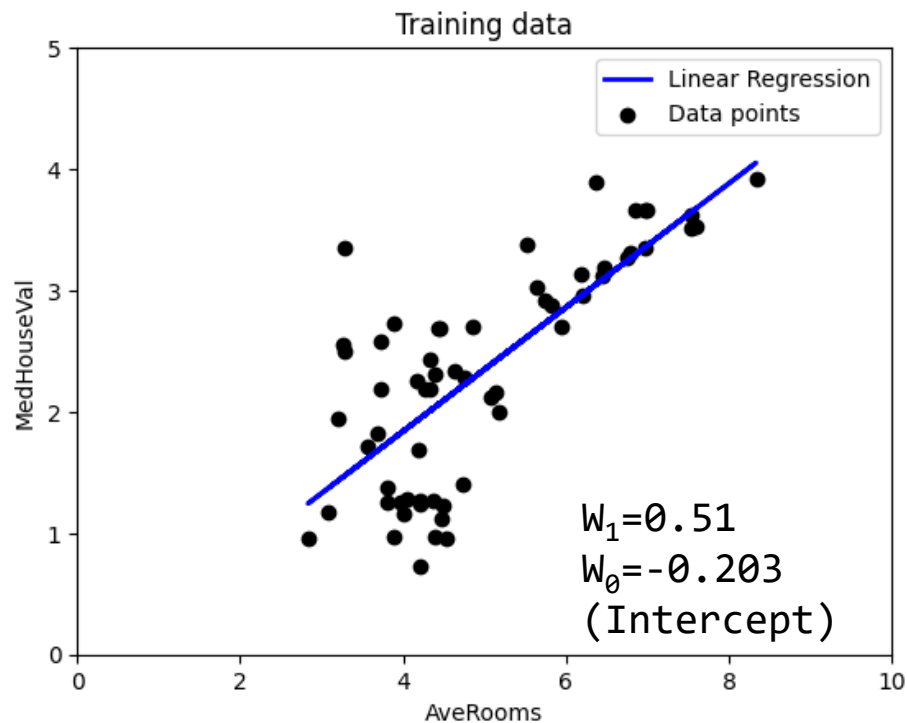
$$X = \begin{bmatrix} 1 & 8.3252 & -122.23 \\ 1 & 8.3014 & -122.22 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ 1 & 3.8462 & -122.25 \end{bmatrix}$$

# Linear Regression: Example

California housing dataset (100 data samples were taken)

Independent variables- AveRooms

Dependent variable is median house value



# Linear Regression: Example

---

Independent variables- AveRooms, Dependent variable is median house value

- RSS 21.89 RSE 0.61 TSS 49.85  $R^2 = 0.56$
- standard errors for intercept and  $w_1$ : 0.089 0.0034
- t-statistic for intercept and  $w_1$ : -0.68 8.75
- pvalue for intercept and  $w_1$ : 0.25  $1.69e^{-12}$

P-value for intercept = 0.25

not statistically significant at conventional significance levels (such as 0.05 or 0.01)

There's no strong evidence to suggest that the constant term in the model significantly affects the response variable. This could imply that the relationship between the predictor and the response starts from or passes through the origin (0,0)

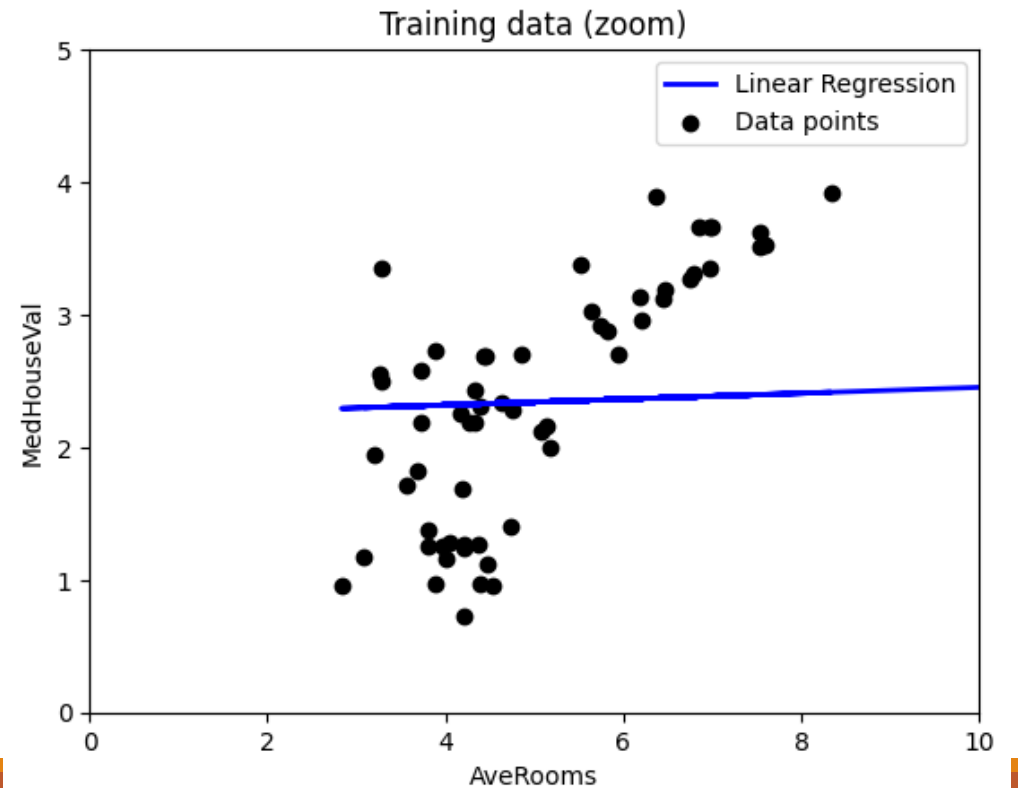
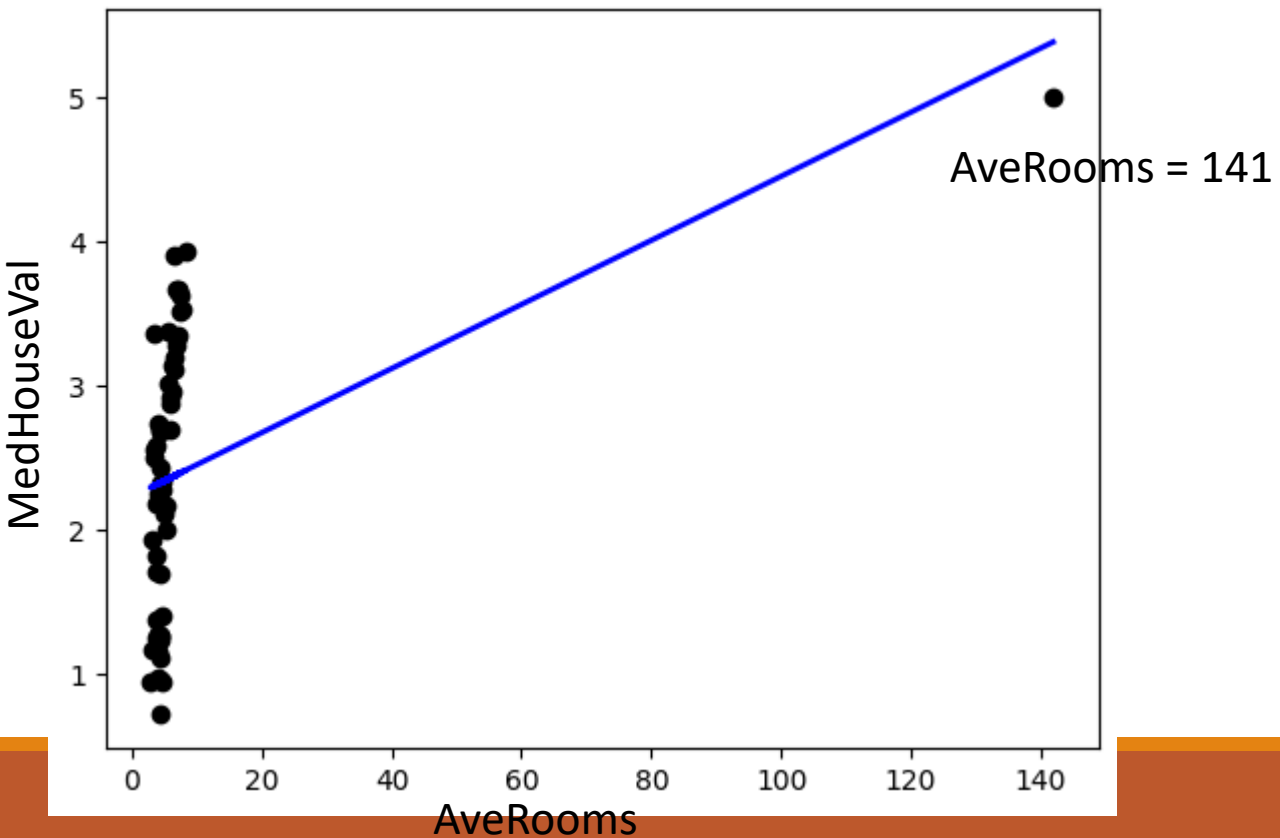
P-value for  $w_1 = 1.69e^{-12}$

coefficient  $w_1$  is likely to be statistically significant and suggests that there is a meaningful linear relationship between the predictor variable ( $w_1$ ) and the response variable

# Linear Regression: Example

Independent variables- AveRooms, Dependent variable is median house value

Impact of outliers





# Linear Regression: Example

Independent variables

MedInc HouseAge AveRooms AveBedrms Population AveOccup Latitude Longitude

Dependent variable

MedHouseVal

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude
Coefficients:	4.52e <sup>-1</sup>	9.44e <sup>-3</sup>	-1.30e <sup>-1</sup>	8.24e <sup>-1</sup>	-5.84e <sup>-6</sup>	-7.62e <sup>-3</sup>	-4.06e <sup>-1</sup>	-4.187e <sup>-1</sup>

	coef	std err	t	P> t	[0.025	0.975]
const	-35.6684	0.849	-42.006	0.000	-37.333	-34.004
x1	0.4527	0.006	81.204	0.000	0.442	0.464
x2	0.0094	0.001	16.493	0.000	0.008	0.011
x3	-0.1302	0.008	-16.185	0.000	-0.146	-0.114
x4	0.8249	0.041	20.252	0.000	0.745	0.905
x5	-5.847e-06	6.15e-06	-0.951	0.341	-1.79e-05	6.2e-06
x6	-0.0076	0.001	-6.442	0.000	-0.010	-0.005
x7	-0.4063	0.009	-43.858	0.000	-0.424	-0.388
x8	-0.4184	0.010	-43.176	0.000	-0.437	-0.399

R-squared: 0.614

Mean squared error: 0.53

A small p-value (usually < 0.05) suggests that the feature is likely to have a significant impact on the target variable. A large p-value suggests that the feature might not have a statistically significant impact on the target variable.

# Ridge Regression

---

$$J(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^N (y_i - \mathbf{w}^T \mathbf{x}_i)^2 + \lambda \|\mathbf{w}\|_2^2$$

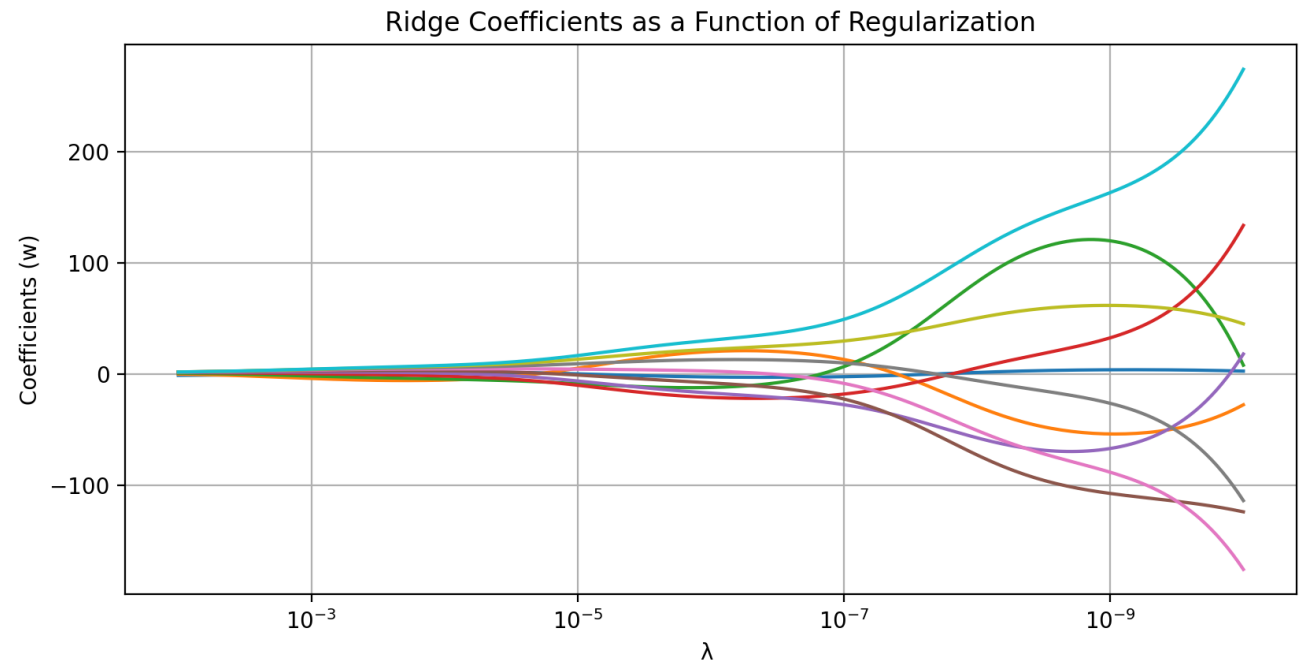
L2 regularization penalty (shrinkage penalty)

- $\|\mathbf{w}\|_2^2 = \sum_j w_j^2 = \mathbf{w}^T \mathbf{w}$  = the squared two-norm.
- $\lambda \geq 0$ , is a complexity penalty (hyper parameter, need to tune).
- Ridge regression also known as Tikhonov regularization, penalized least squares  $L_2$  regularization, or weight decay.
- Address multicollinearity (high correlation between predictor (independent) variables)
- Avoid overfitting
- Regularization discourages the model from assigning overly large coefficients to predictor variables

$$\hat{\mathbf{w}}_{\text{ridge}} = (\lambda \mathbf{I}_D + \mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

# Ridge Regression: Impact of $\lambda$

- $J(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^N (y_i - \mathbf{w}^T \mathbf{x}_i)^2 + \lambda \|\mathbf{w}\|_2^2$
- When  $\lambda$  is very large (coefficients  $\mathbf{w} \rightarrow 0$ ).
- When  $\lambda$  tends toward zero, Ordinary least square  $\rightarrow$  coefficients exhibit big oscillations.
- Need to tune  $\lambda$  to balance



# Polynomial Regression

---

➤ Model: Relationship between the independent variable(s) and the dependent variable is modeled as an  $n^{\text{th}}$ -degree polynomial\*.

➤  $y = \mathbf{w}^T \phi(\mathbf{x})$  and for single variable  $\phi(\mathbf{x}) = [1, x, x^2, \dots, x^d]$

➤ For two feature  $d = 2$ ,  $\phi(\mathbf{x}) = [1, x_1, x_2, x_1^2, x_1 x_2, x_2^2]$

➤ Example  $y = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_1 x_2 + w_5 x_2^2$   
 $y = w_0 + w_1 z_1 + w_2 z_2 + w_3 z_3 + w_4 z_4 + w_5 z_5$

➤ Cost function

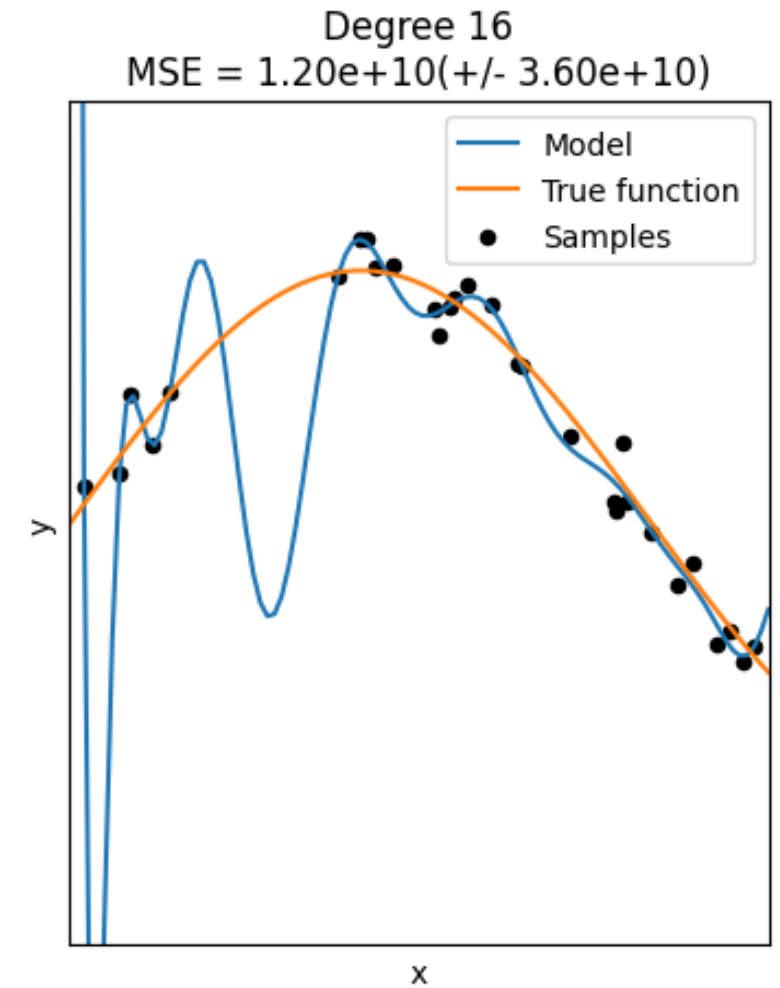
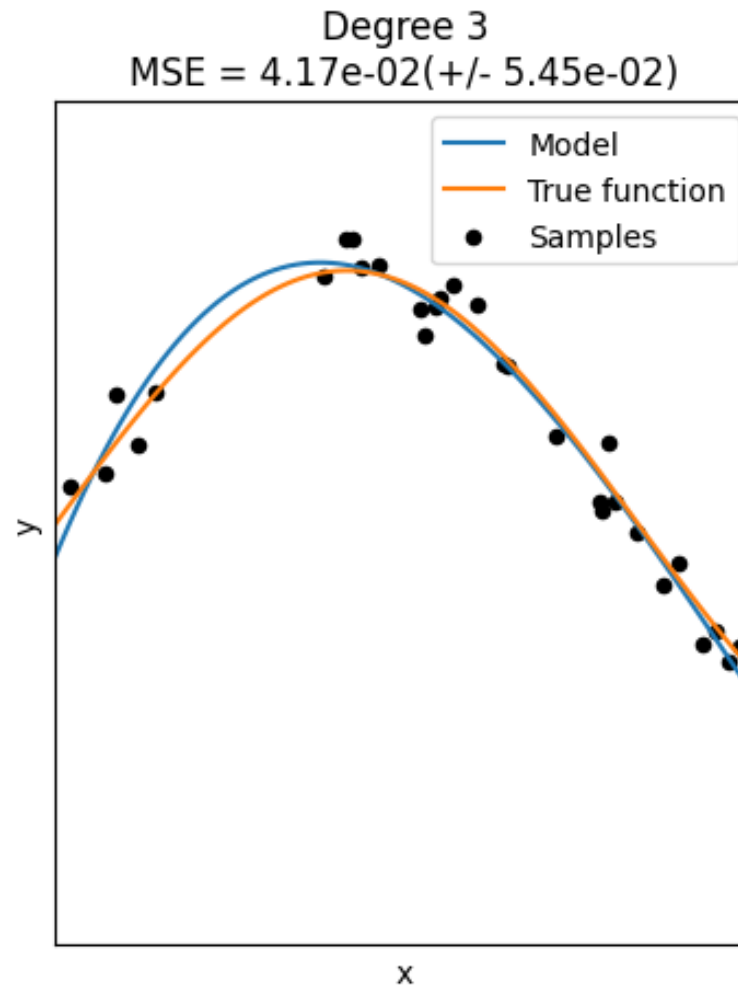
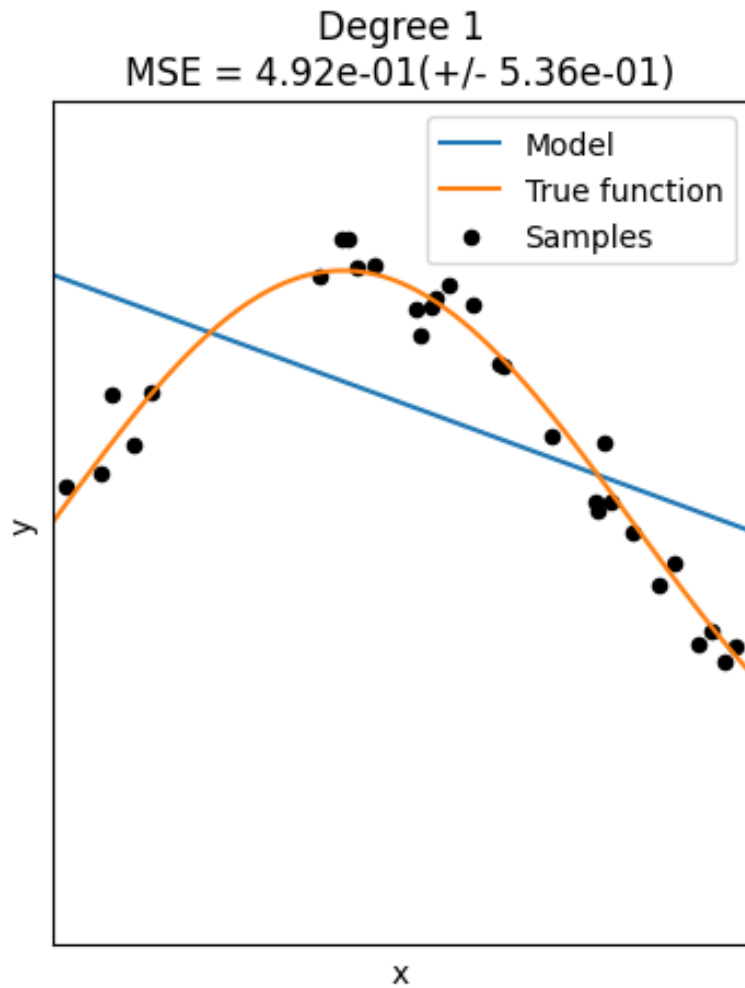
$$J(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^N (y_i - \mathbf{w}^T \phi(\mathbf{x}))^2$$

➤ Polynomial **Ridge** Regression

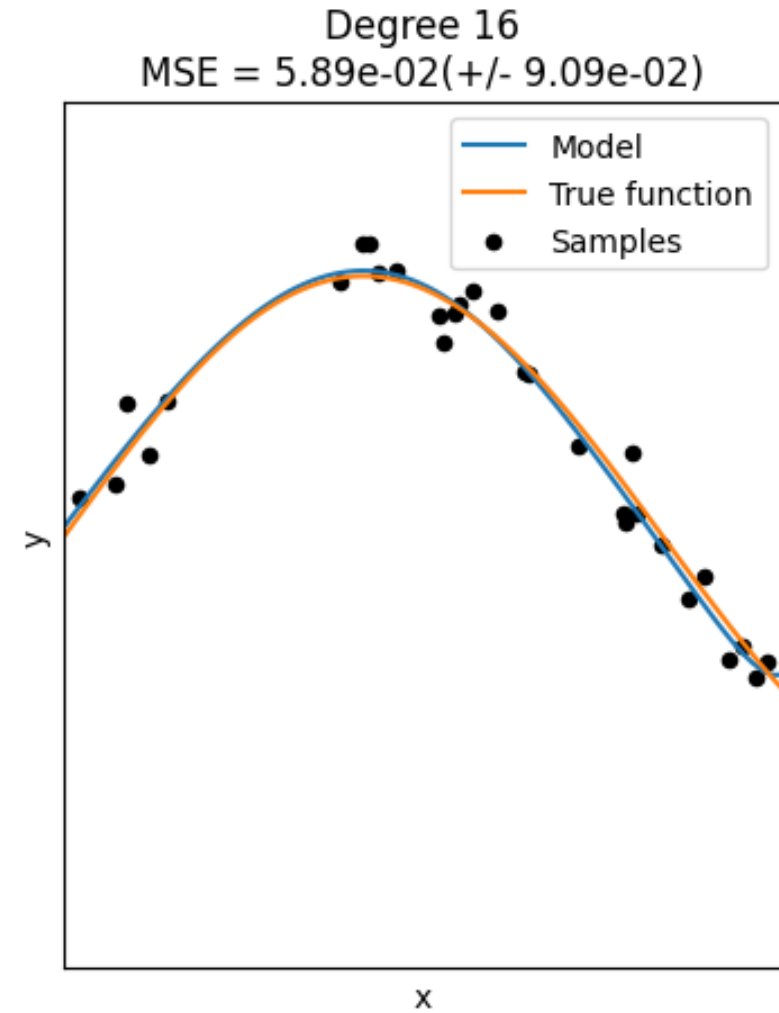
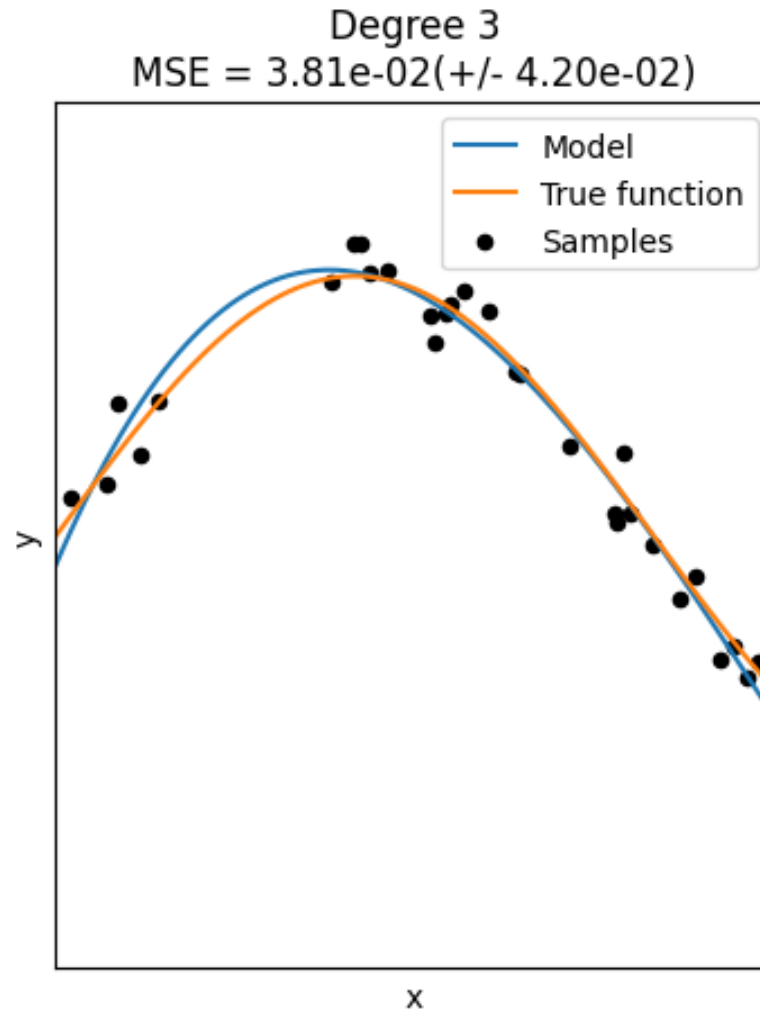
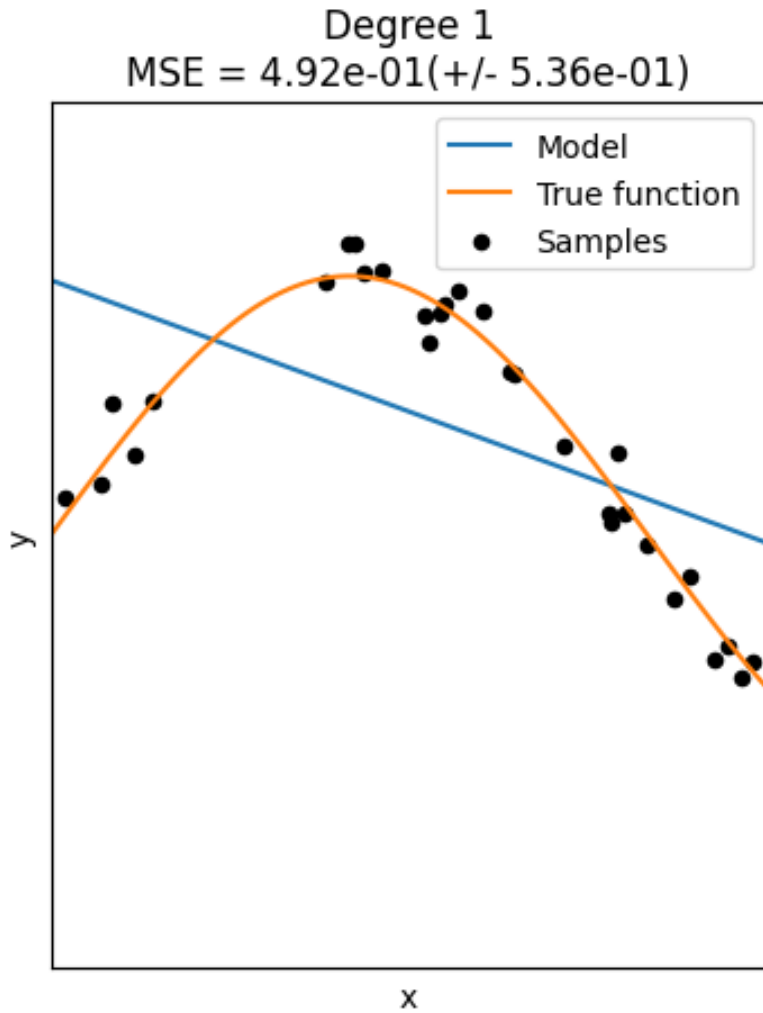
$$J(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^N (y_i - \mathbf{w}^T \phi(\mathbf{x}))^2 + \lambda \|\mathbf{w}\|_2^2$$

\* Still a linear model with respect to weights. Nonlinear model when  $x_1^{\wedge} w_1$

# Polynomial Regression



# Polynomial Ridge Regression



# Lasso

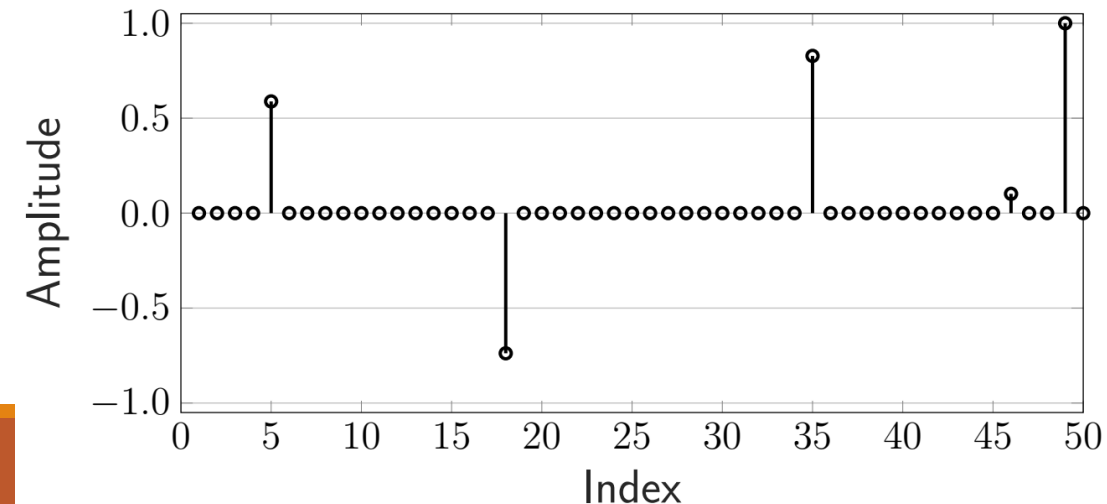
➤ Least Absolute Shrinkage and Selection Operator

➤ Cost function  $J(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^N (y_i - \mathbf{w}^T \mathbf{x})^2 + \lambda \|\mathbf{w}\|_1$  L1 regularization penalty

➤ Select a subset of important features while reducing the coefficients of less important features to nearly zero (Encourages sparsity)

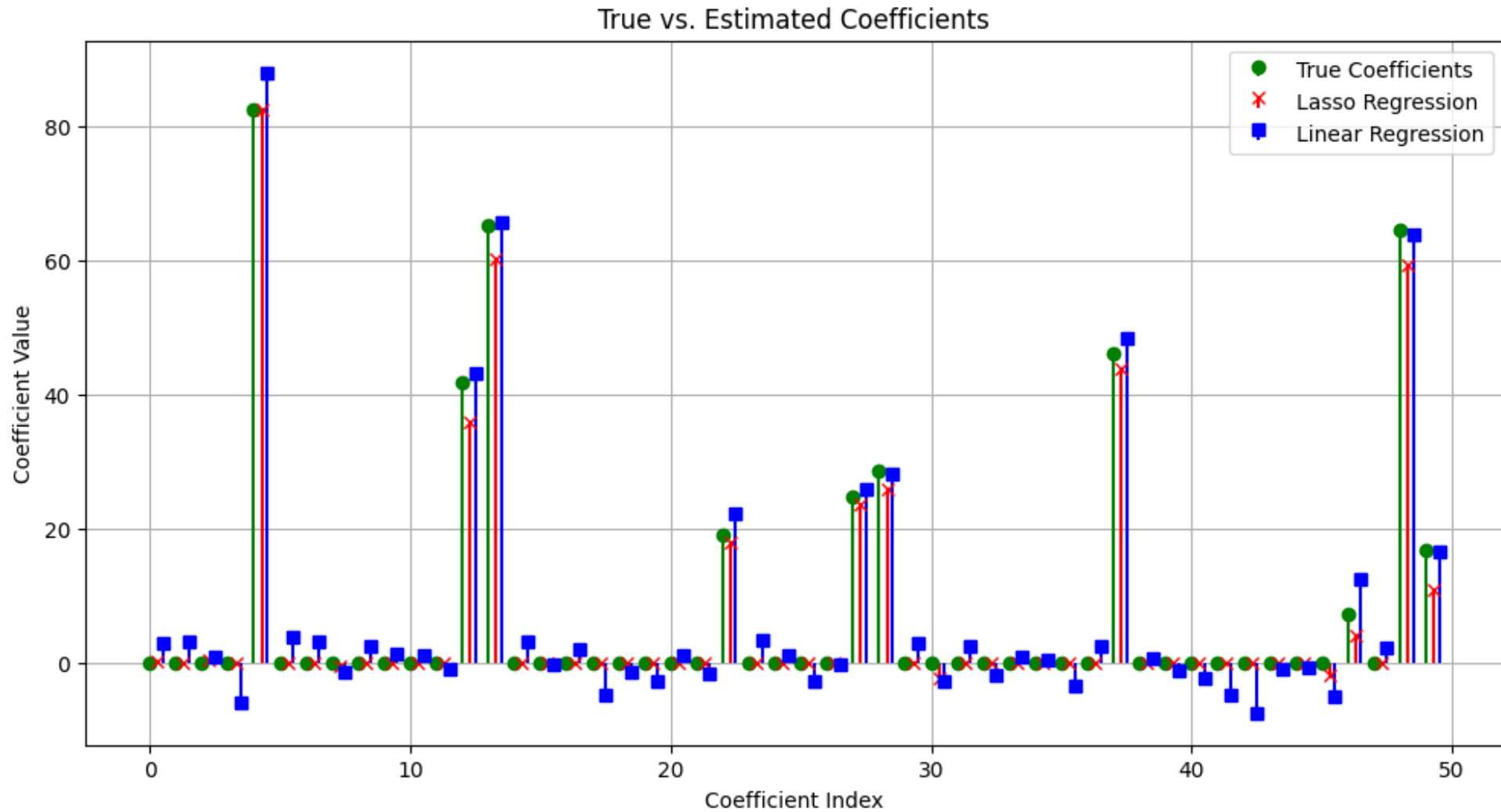
➤ Useful for feature selection or dimensionality reduction

➤ Lasso plays important role in compressed sensing



[Introduction to Sparsity in Signal Processing \(nyu.edu\)](#)

# Lasso



Number of non-zero coefficients (Linear Regression): 50

Number of non-zero coefficients (Lasso Regression): 16

True number of non-zero coefficients = 10 and total number of coefficients = 50



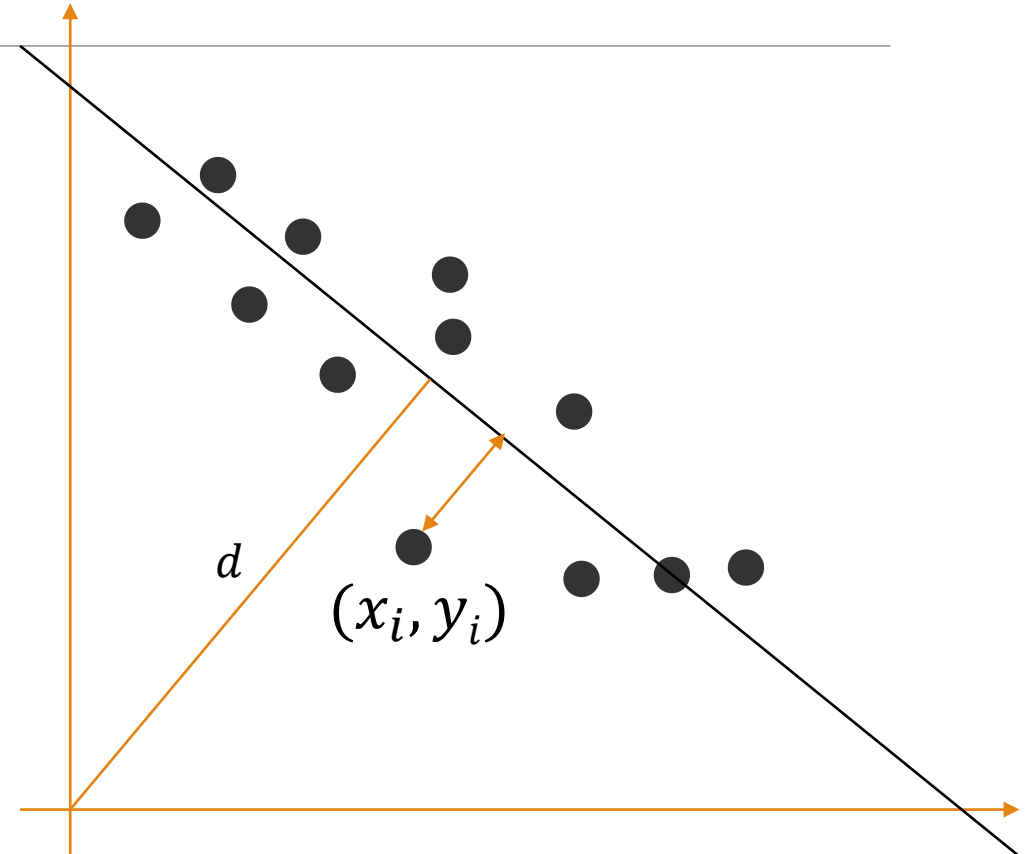
# Total least squares (TLS)

- Equation of a line can be written as  $ax + by = d$
- $a^2 + b^2 = 1$ , where  $(a, b)$  is the unit normal to the line
- Distance between line and the origin is given as  $d$
- Extension to 3d,  $ax + by + cz = d$  and  $a^2 + b^2 + c^2 = 1$
- Distance (perpendicular) from point to line is  $\frac{|ax+by-d|}{\sqrt{a^2+b^2}}$
- Loss function:

$$E = \sum_{i=1}^n (ax_i + by_i - d)^2$$

$$\hat{a}, \hat{b}, \hat{d} = \arg \min_{a, b, d} E \quad \times$$

$$\begin{aligned} \hat{a}, \hat{b}, \hat{d} = \arg \min_{a, b, d} E \\ \text{such that } a^2 + b^2 = 1 \end{aligned}$$



# Total least squares (TLS)

---

➤ Solve for  $d \rightarrow \frac{\partial E}{\partial d} = 0$

$$E = \sum_{i=1}^n (ax_i + by_i - d)^2$$

$$\sum_{i=1}^n 2(ax_i + by_i - d)(-1) = 0$$

$$\sum_{i=1}^n (-ax_i - by_i + d) = 0$$

$$n d = \sum_{i=1}^n (ax_i + by_i)$$

$$d = \frac{a}{n} \sum_{i=1}^n x_i + \frac{b}{n} \sum_{i=1}^n y_i = a\bar{x} + b\bar{y}$$

$$E = \sum_{i=1}^n (a(x_i - \bar{x}) + b(y_i - \bar{y}))^2 = \left\| \begin{bmatrix} x_1 - \bar{x} & y_1 - \bar{y} \\ \vdots & \vdots \\ x_n - \bar{x} & y_n - \bar{y} \end{bmatrix} \begin{pmatrix} a \\ b \end{pmatrix} \right\|^2 = \|\mathbf{X}\mathbf{u}\|^2$$

# Total least squares (TLS)

---

- Optimization problem minimizes  $\|X\mathbf{u}\|^2$  subject to  $\|\mathbf{u}\|^2 = 1$
- Lagrangian function  $E(\mathbf{u}, \lambda) = \|X\mathbf{u}\|^2 + \lambda (\|\mathbf{u}\|^2 - 1)$  and  $\lambda$  is Lagrange multiplier
- $\frac{\partial E}{\partial \mathbf{u}} = 0, \Rightarrow 2 X^T(X \mathbf{u}) + \lambda \mathbf{u} = 0 \Rightarrow (X^T X) \mathbf{u} = -\lambda \mathbf{u}$
- $(X^T X) \mathbf{u} = -\lambda \mathbf{u}$  Some thing familiar?
- $\mathbf{u}$  is the eigenvector of  $(X^T X)$ , if  $(X^T X)$  is  $p$  by  $p$  vector and there can be  $p$  number of eigen vectors and eigen values ( $\lambda$ )
- Which one should I pick?
- $\frac{\partial E}{\partial \lambda} = 0 \Rightarrow \|\mathbf{u}\|^2 - 1 \Rightarrow \|\mathbf{u}\|^2 = 1$  already known.
- $(X^T X) \mathbf{u} = -\lambda \mathbf{u}$  multiply both sides by  $\mathbf{u}^T$
- $\mathbf{u}^T (X^T X) \mathbf{u} = -\lambda \mathbf{u}^T \mathbf{u}$
- $(\mathbf{u}^T X^T) (X \mathbf{u}) = -\lambda$  any insight for picking value to value for  $\lambda$  ?

# Total least squares (TLS)

---

➤  $(\mathbf{u}^T \mathbf{X}^T) (\mathbf{X} \mathbf{u}) = -\lambda$  any insight for picking value to value for  $\lambda$  ?

➤  $\|\mathbf{X}\mathbf{u}\|^2 = -\lambda$ , we need to minimize the  $\|\mathbf{X}\mathbf{u}\|^2$ .

➤ Pick a smallest eigen value.

➤ Solution: eigenvector of  $(\mathbf{X}^T \mathbf{X})$  associated with the smallest eigenvalue.

e. g.,  $\begin{bmatrix} 0.4 & 0.68 \\ 0.05 & 0.39 \end{bmatrix}$  for this matrix eigenvalues and eigenvectors are  $[0.58 \quad 0.22]$  and  $\begin{bmatrix} 0.96 & -0.96 \\ 0.25 & 0.26 \end{bmatrix}$

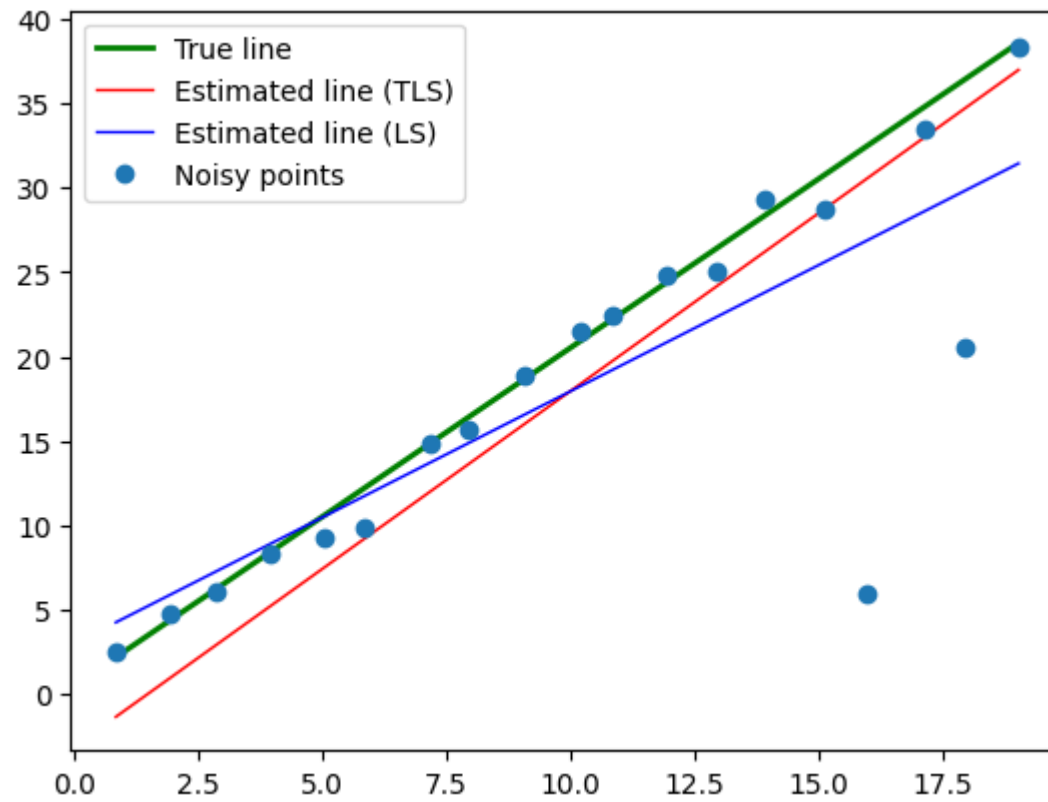
➤ LS assume that measurement errors primarily affect the dependent variable (the y-values).

➤ TLS consider both dependent and independent variables have measurement errors.

➤ TLS is less sensitive to outliers compared to traditional least squares.

# TLS and LS

---



# Self Study

---

## ➤ **RANSAC** (RANdom SAmple Consensus)

- Fischler, Martin A., and Robert C. Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography." *Communications of the ACM* 24.6 (1981): 381-395.
- [https://scikit-learn.org/stable/modules/generated/sklearn.linear\\_model.RANSACRegressor.html](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.RANSACRegressor.html)
- Robust linear model estimation using RANSAC — scikit-learn 1.3.0 documentation



---

Thank You

Q & A