Applied Machine Learning

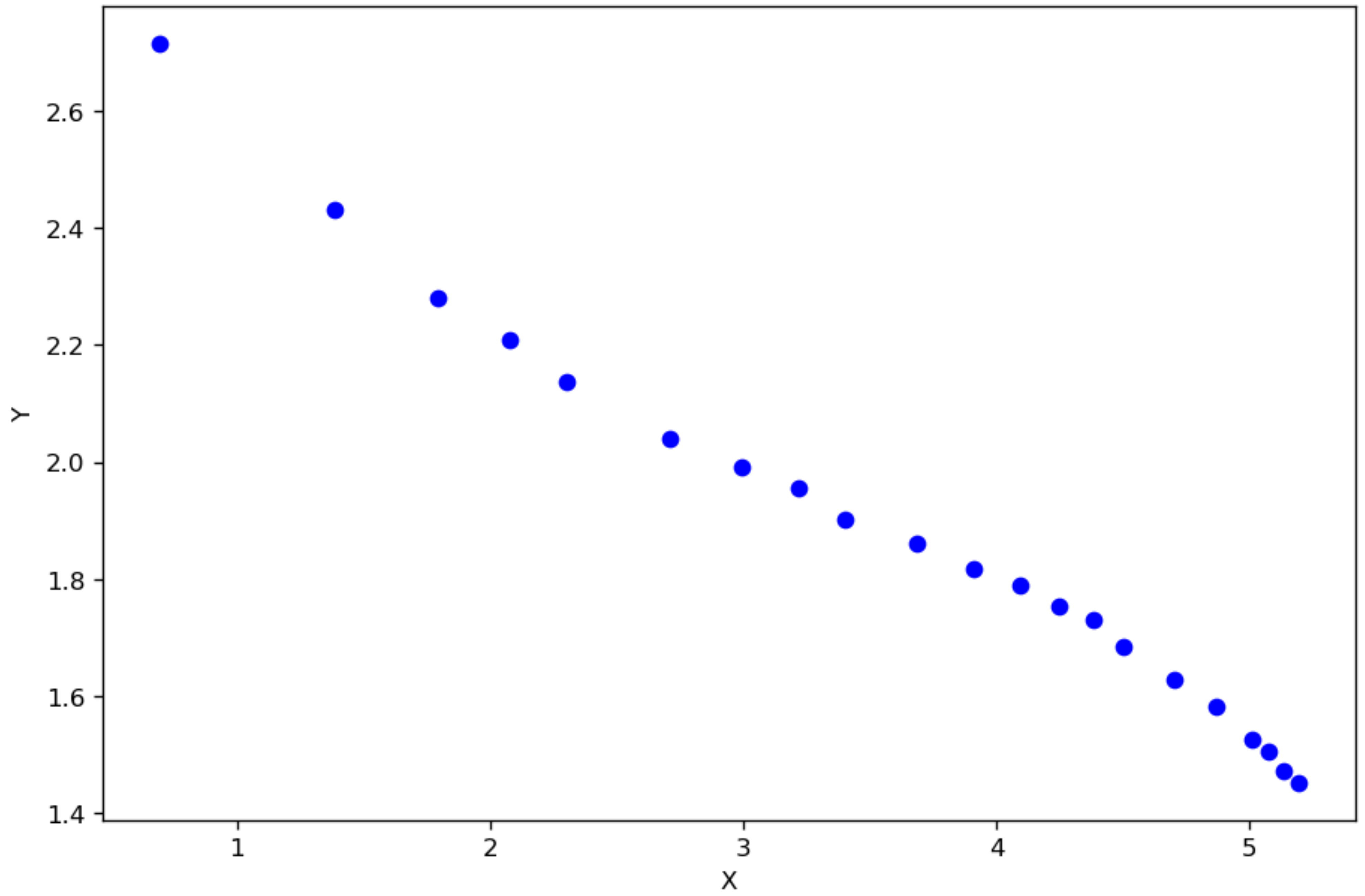
Linear Regression

Linear Regression

- Overview
- Linear regression
- Regression coefficients

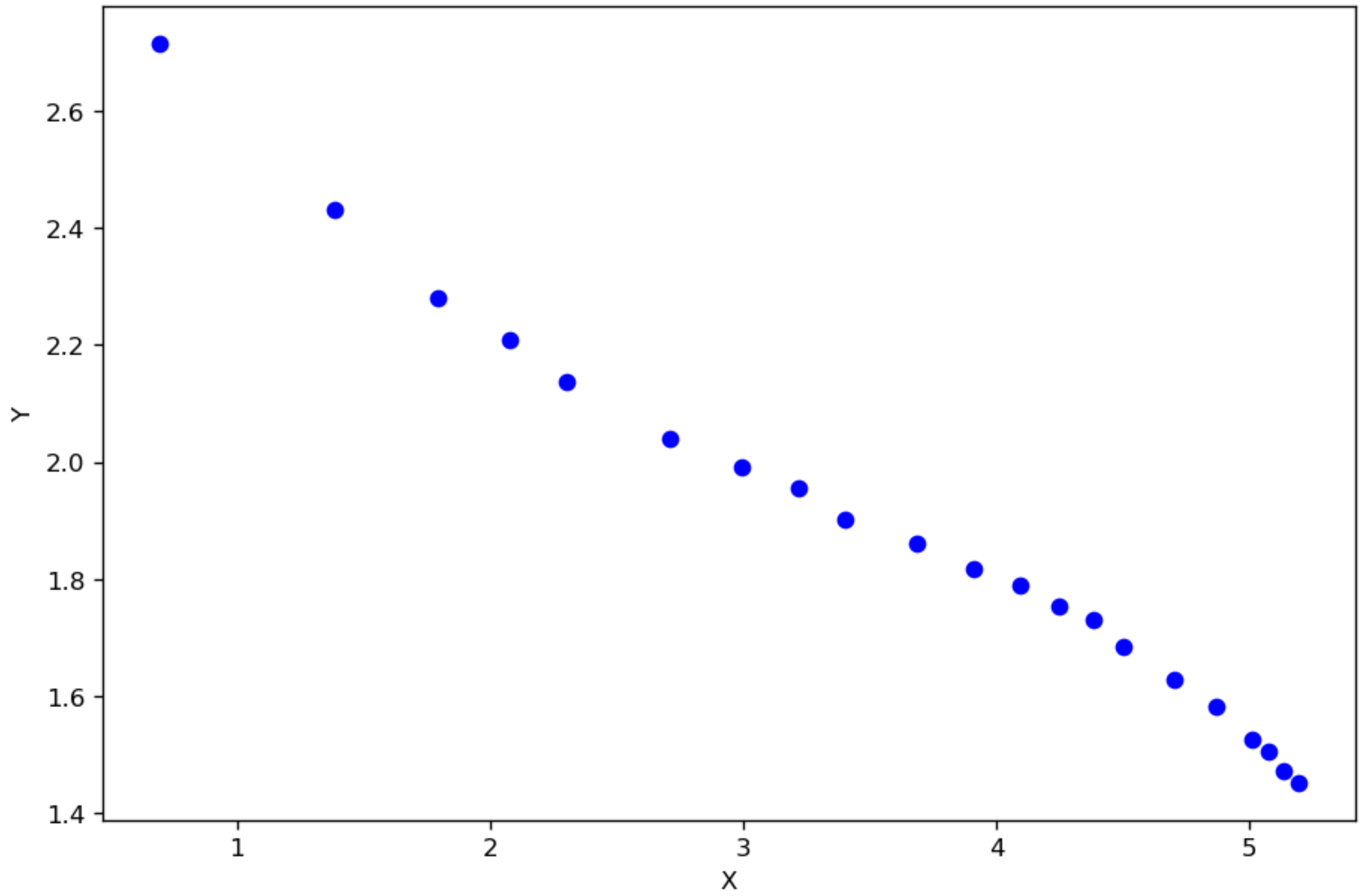
Linear Regression - Overview

- Linear classifier
    - N pairs of $(\mathbf{x}_i, y_i)$ items
        - $\mathbf{x}_i$: feature vector
        - y_i: numerical value of function evaluated at $\mathbf{x}_i$
- Goal
    - find a linear function that models the dataset items
- Regressing dependent variable against explanatory variable
    - Given a new feature vector $\mathbf{x}$: explanatory variables
    - Predict numerical value of y: dependent variable



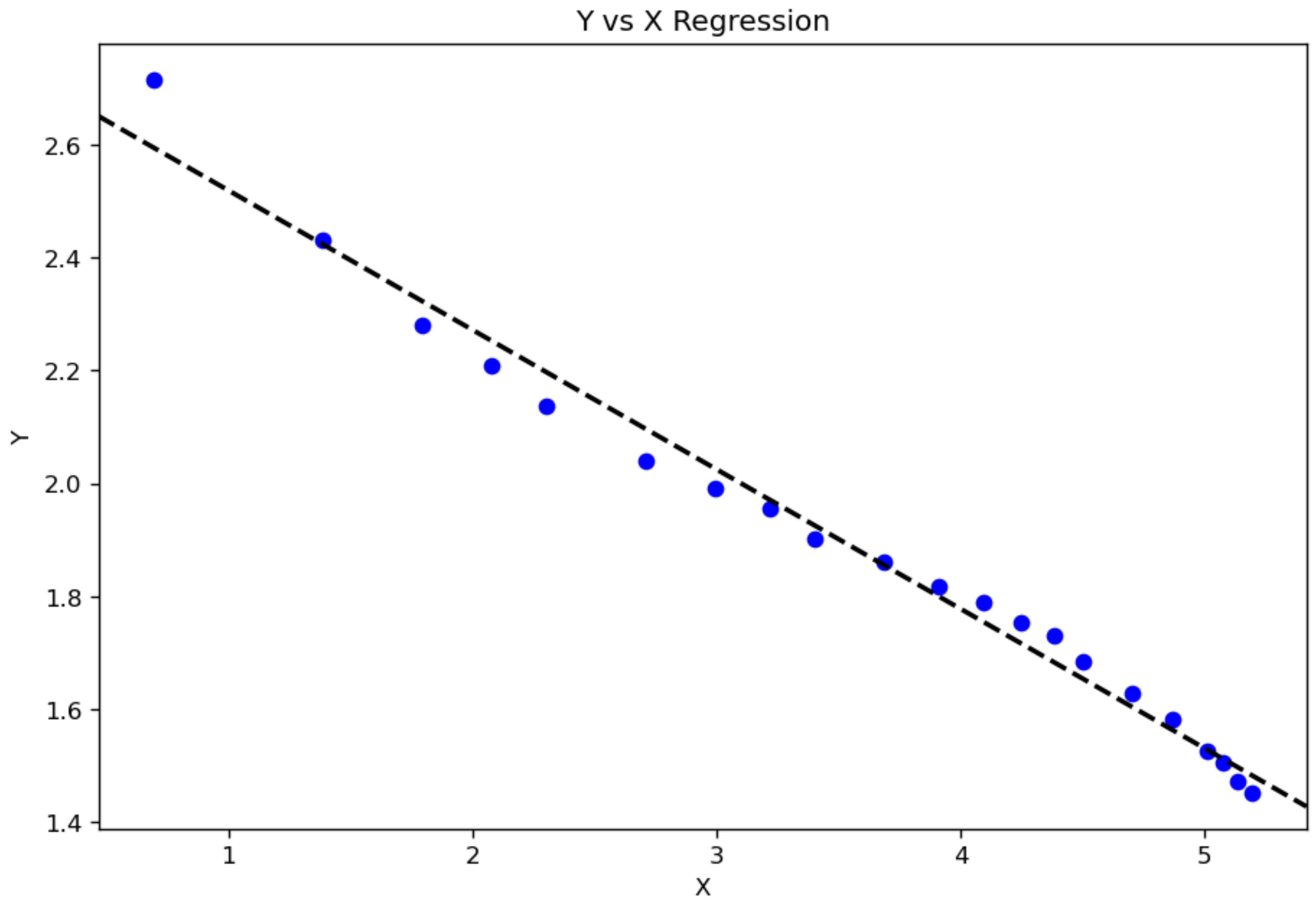
Linear Regression - Overview

- Regression
  - Given a new feature vector $\mathbf{x}$
  - Predict numerical value of y
- Comparing trends in data
  - Identifying unexpected biases



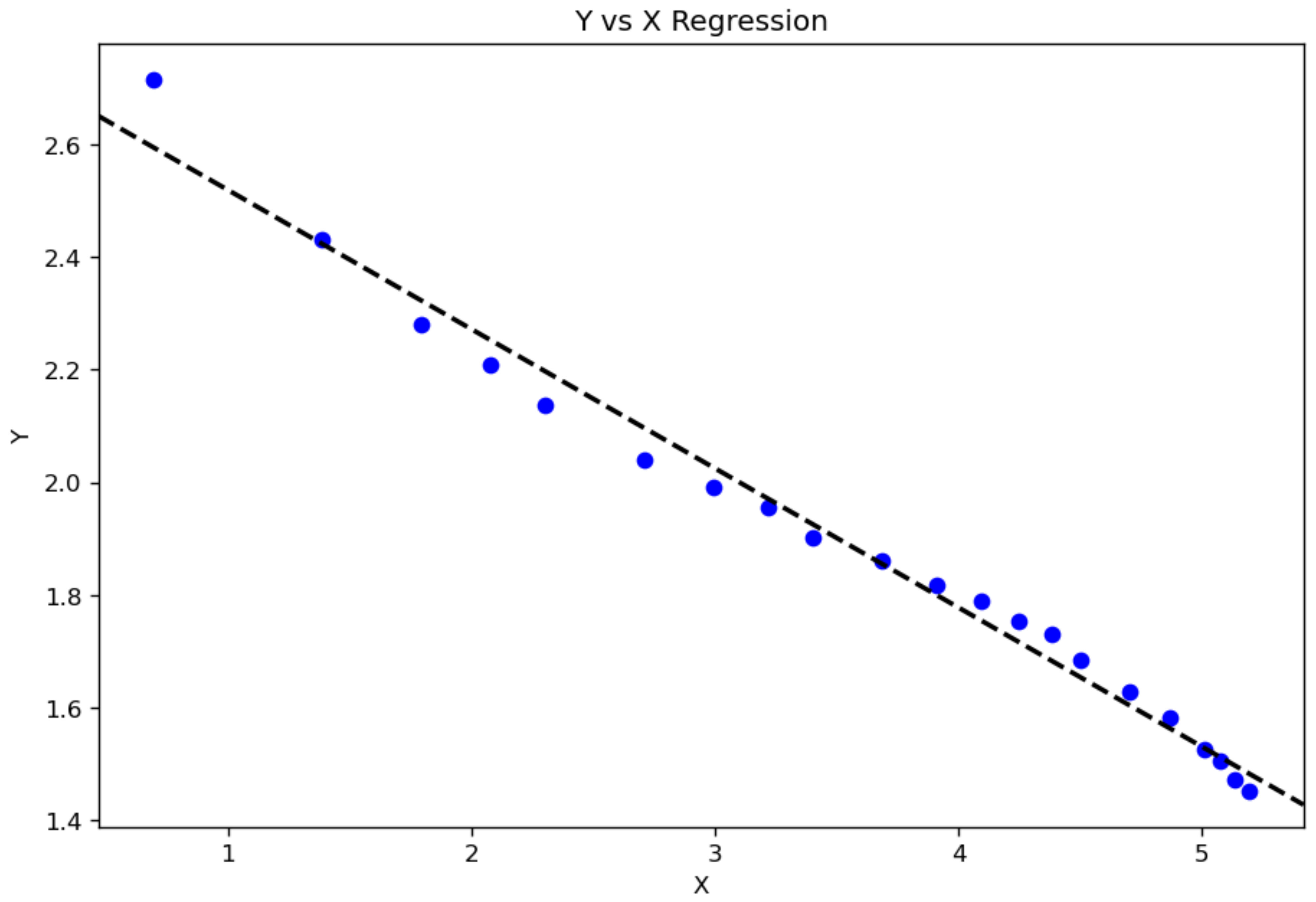
Linear Regression - Overview

- Linear regression for one feature:
  - $ax + b$
  - Given a new feature value x
  - Predict numerical value of y
- $residual = y_{actual} - y_{predicted}$



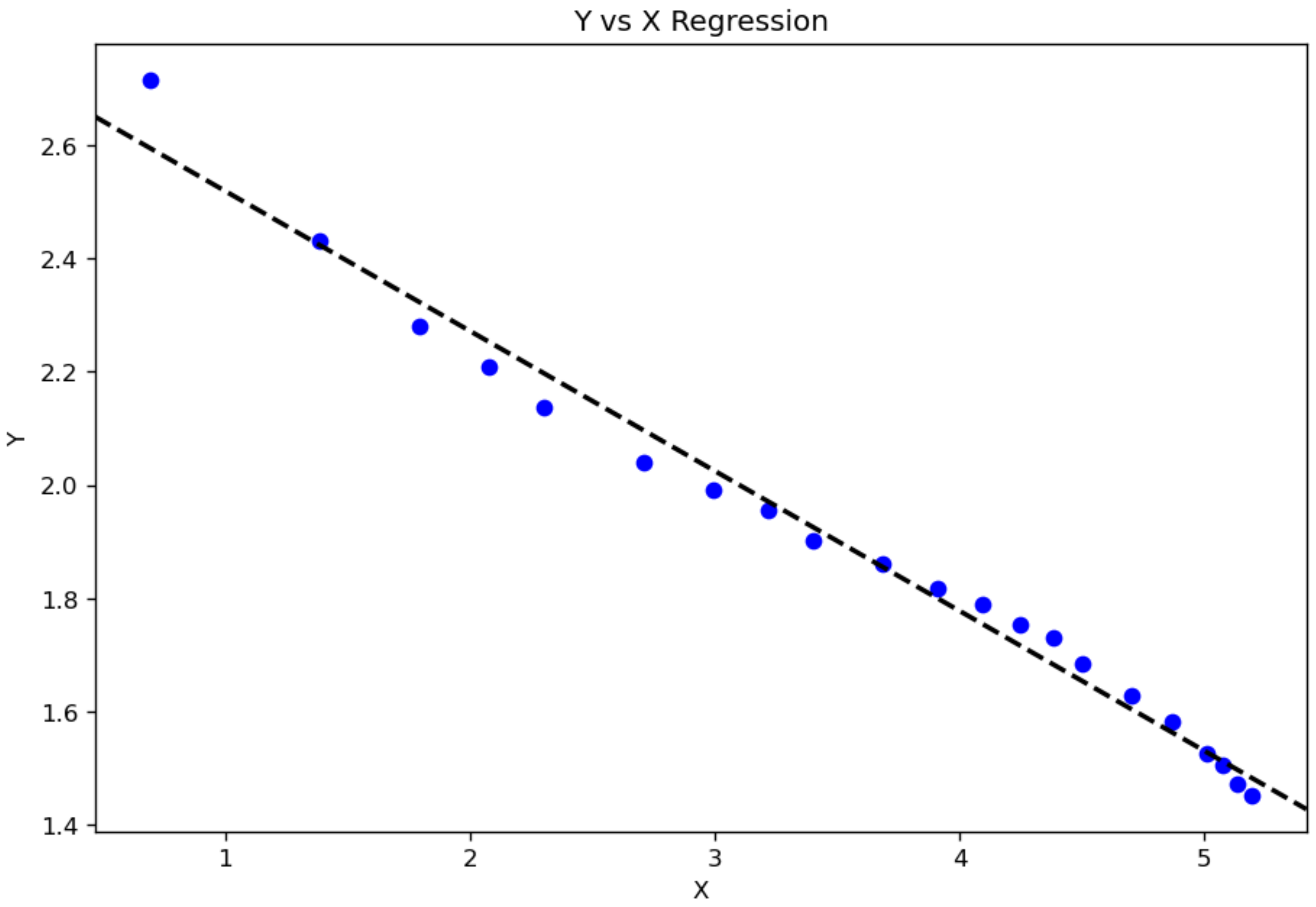
Linear Regression - Overview

- y_i may be different for two items with the same feature vector $\mathbf{x}_i$
    - randomness
    - y may not be a function of $\mathbf{x}$
- Random variables Y and $\mathbf{X}$
    - y_i is sample from $P(Y | X = \mathbf{x_i})$



Linear Regression

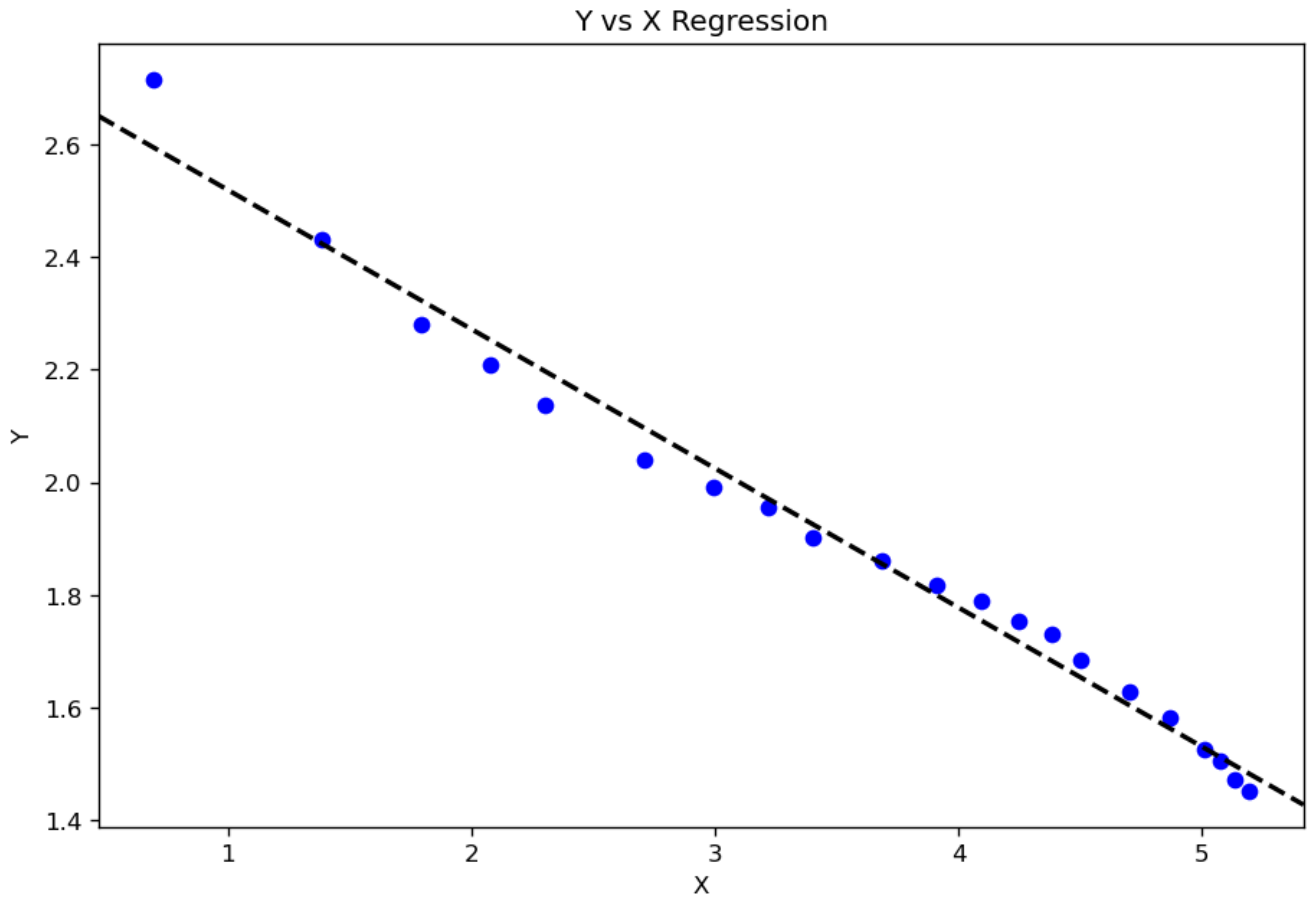
- Regression model
    - $y = \mathbf{x}^\top \beta + \xi$
    - β: vector of coefficients
    - ξ: random normal noise with zero mean
    - y-intercept: add one feature value of 1 in $\mathbf{x}$
    - $\mathbf{x} = \begin{bmatrix} x_n \\ \vdots \\ x_1 \\ 1 \end{bmatrix}$



- Prediction: given a new $\mathbf{x}^*$, predict y^*

Linear Regression - 1 variable

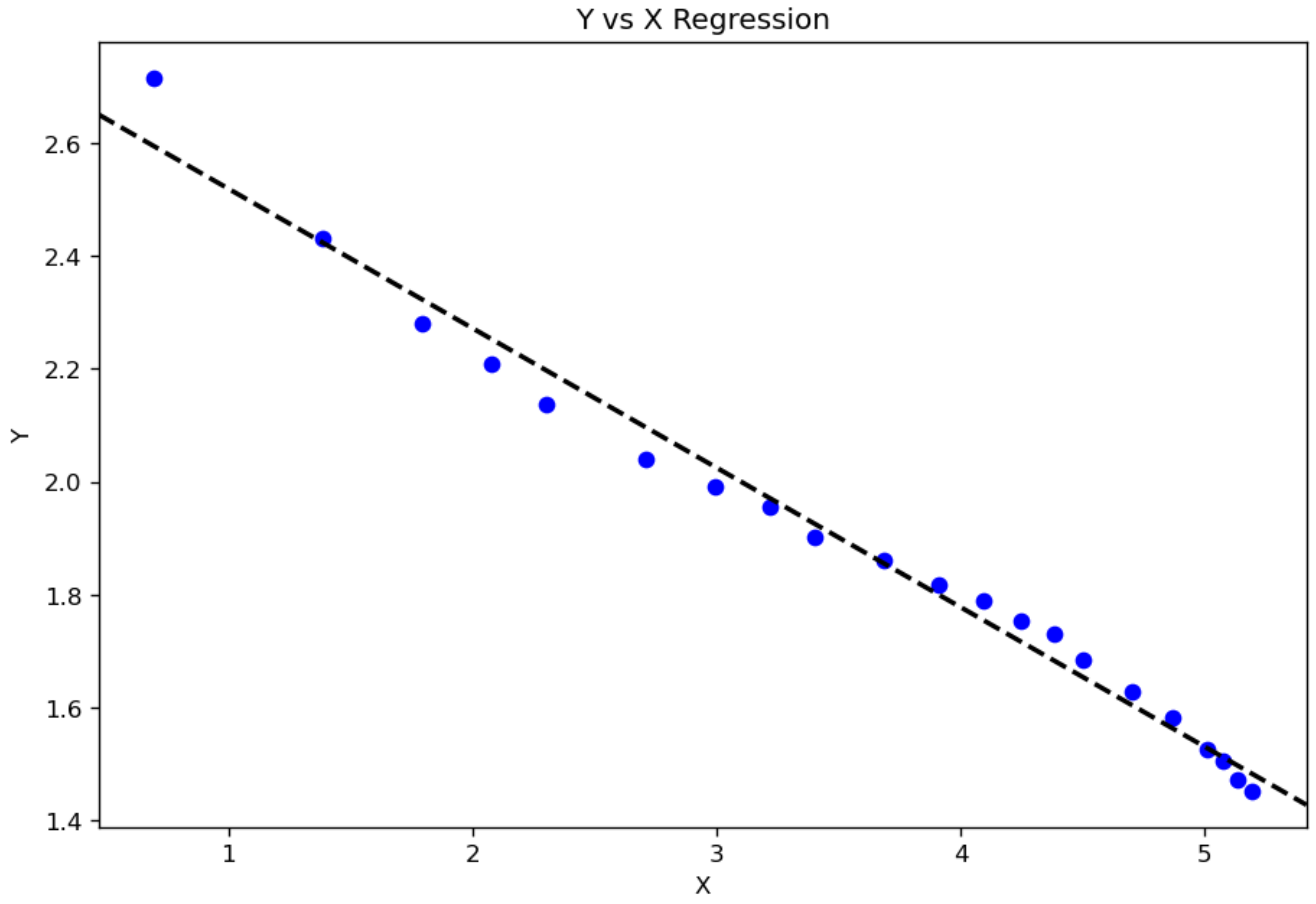
- Regression model
  - $y = \mathbf{x}^\top \beta + \xi$
    - $\mathbf{x}^* = \begin{bmatrix} x^* \\ 1 \end{bmatrix}$
    - $y = \beta_1 x^* + \beta_2$
      - β_1: slope
      - β_2: y-intercept



$\beta_1 \approx -0.24, \beta_2 \approx 2.65$

Linear Regression - finding β

- Regression model
  - $y = \mathbf{x}^\top \beta + \xi$
- Goal:
  - minimize ξ
- Two ways to get to the same minimization goal
  - Direct
  - Probabilistic



Linear Regression - finding β

- Regression model: $y = \mathbf{x}^\top \beta + \xi$
- Goal: minimize $\xi_i = y_i - \mathbf{x}_i^\top \beta$
- mean squared error to minimize: $\frac{1}{N}\left(\sum_i (y_i - \mathbf{x}_i^\top \beta)^2 \right)$

Linear Regression - finding β

- Regression model: $y = \mathbf{x}^\top \beta + \xi$
- Goal: minimize $\xi_i = y_i - \mathbf{x}_i^\top \beta$
  - ξ normal random variable with zero mean
  - $P(y \mid \mathbf{x}, \beta)$ mean: $\mathbf{x}^\top \beta$; variance: σ
- Maximize log-likelihood (min negative log-likelihood)
  - $$\begin{aligned} \log \mathscr{L}(\beta) &= \sum_i \log P(y \mid \mathbf{x}, \beta) \\ &= -\frac{1}{2\sigma^2} \sum_i (y_i - \mathbf{x}_i^\top \beta)^2 - \frac{N}{2} \log(2\pi\sigma^2) \end{aligned}$$
- mean squared error to minimize: $\frac{1}{N} \left(\sum_i (y_i - \mathbf{x}_i^\top \beta)^2 \right)$

Linear Regression - finding β

- $\mathbf{y}_{[N\times 1]} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} \qquad \mathbf{X}_{[N\times(k+1)]} = \begin{bmatrix} \mathbf{x}_1^\top \\ \vdots \\ \mathbf{x}_N^\top \end{bmatrix}$
- mean squared error to minimize
  - $\frac{1}{N}\left(\sum_i (y_i - \mathbf{x}_i^\top \beta)^2\right) = \frac{1}{N}(\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta)$
- Differentiating and equating to 0:

$$\begin{aligned} \mathbf{X}^\top\mathbf{X}\beta - \mathbf{X}^\top\mathbf{y} &= 0 \\ \beta &= (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y} \end{aligned}$$

  - direct if $(\mathbf{X}^\top\mathbf{X})^{-1}$ has full rank

Linear Regression

- Overview
- Linear regression
- Regression coefficients

Applied Machine Learning

Linear Regression