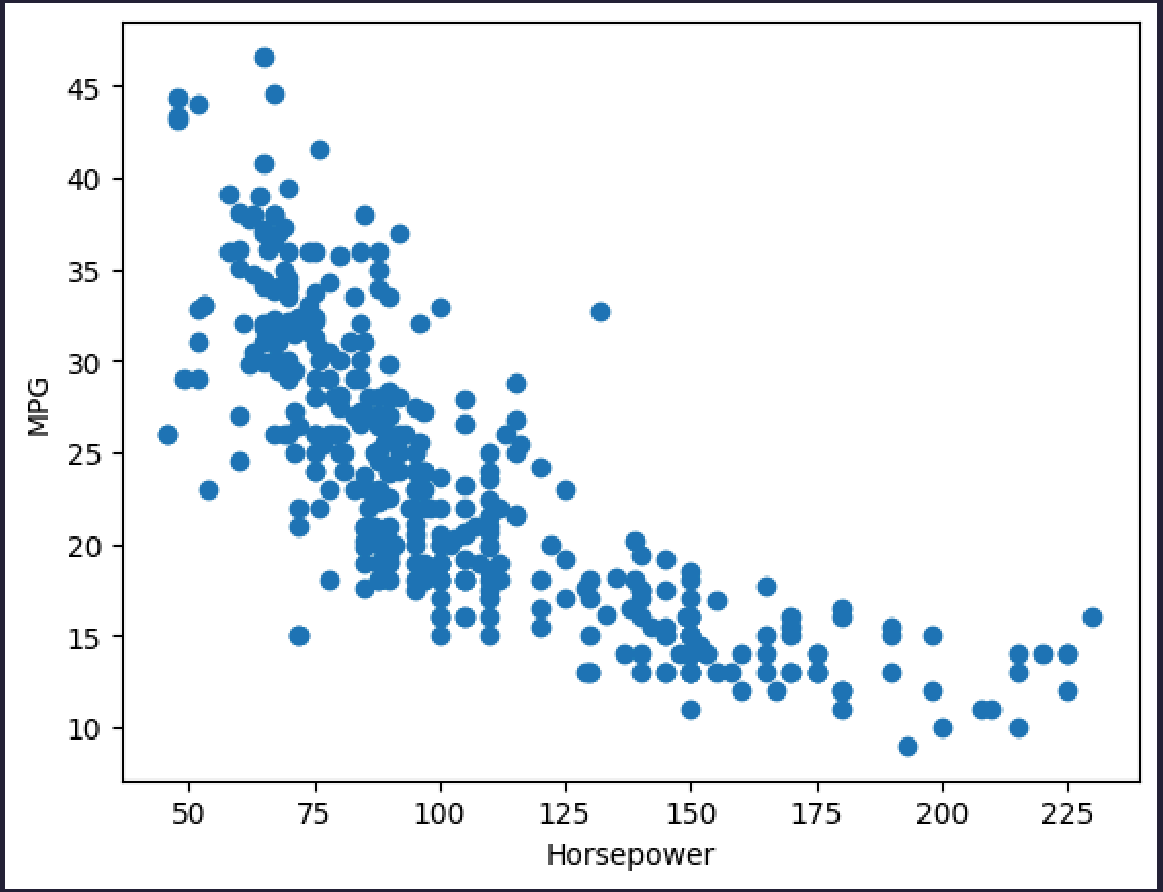
Regression 2

Multiple features

	mpg (y)	weight (x1)	horsepower (x2)
0	18.0	3504	130
1	15.0	3693	165
2	18.0	3436	150
3	16.0	3433	150
4	17.0	3449	140



Linear regression with single feature

model

$$f(x) = wx + b$$

parameters

$$w, b$$

cost function

$$J(w, b)$$

goal

find w and b that minimize $J(w, b)$

Multivariable (multiple) regression

model

$$f(x) = w_1x_1 + w_2x_2 + \ldots + w_nx_n + b$$

parameters

$$w_1, w_2, \ldots, w_n, b$$

cost function

$$J(w_1, w_2, \ldots, w_n, b)$$

goal

find $w_1, w_2, \ldots, w_n, b$ that minimize $J(w_1, w_2, \ldots, w_n, b)$

Multivariable (multiple) regression

model

$f(\vec{x}) = \vec{w} \cdot \vec{x} + b$

parameters

$\vec{w}, b$

cost function

$J(\vec{w}, b)$

goal

find $\vec{w}, b$ that minimize $J(\vec{w}, b)$

Basic linear algebra: vector, matrix, transpose

Column vector (M x 1)

$$\vec{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \vec{v}^T = \begin{bmatrix} v_1 & v_2 \end{bmatrix}$$

Row vector (1 x N)

$$\vec{u} = \begin{bmatrix} u_1 & u_2 \end{bmatrix}, \vec{u}^T = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

matrix (M x N)

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}, A^T = \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \\ a_{13} & a_{23} \end{bmatrix}$$

Basic linear algebra: dot product, multiplication

Dot product: element-wise multiplication and sum

$$\vec{u} \cdot \vec{v} = \begin{bmatrix} u_1 & u_2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = u_1 v_1 + u_2 v_2 = \sum_{i=1}^{n} u_i v_i$$

Matrix multiplication: dot products of rows and columns

$$A^T \vec{v} = \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \\ a_{13} & a_{23} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} a_{11} v_1 + a_{21} v_2 \\ a_{12} v_1 + a_{22} v_2 \\ a_{13} v_1 + a_{23} v_2 \end{bmatrix}$$

🤔 Basic linear algebra

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}, \vec{v} = \begin{bmatrix} 5 \\ 6 \end{bmatrix}$$

Dimension of A?

Dimension of $\vec{v}$?

$A^T A$?

$A^T \vec{v}$?

$A\vec{v}$?

regression (one variable, one sample)

$$w = 3$$
$$b = 4$$
$$x = 1$$

$$f(x) = wx + b = 3 * 1 + 4 = 7$$

regression (one variable, multiple samples)

$$w = 3$$
$$b = 4$$
$$\vec{x} = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ x^{(3)} \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

$$f(\vec{x}) = w\vec{x} + b = 3 \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + 4 = \begin{bmatrix} 7 \\ 10 \\ 13 \end{bmatrix}$$

regression (multiple variables, multiple samples)

$$\vec{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

$$b = 4$$

$$X = \begin{bmatrix} \vec{x}_1 & \vec{x}_2 \end{bmatrix} = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} \\ x_1^{(2)} & x_2^{(2)} \\ x_1^{(3)} & x_2^{(3)} \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \end{bmatrix}$$

$$f(X) = X\vec{w} + b = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \end{bmatrix} + 4 = \begin{bmatrix} 11 \\ 16 \\ 21 \end{bmatrix}$$

`predict` for single variable

$$f(\vec{x}) = w\vec{x} + b$$

```python
def predict(x,w,b):
    """
    x: array (n,): n training samples
    w: scalar
    b: scalar

    return: array (n,)

    """
    # array (n,) = scalar * array (n,) + scalar
    return w*x + b
```

`predict` for multiple variables (no vectorization)

$$f(\vec{x}) = w_1\vec{x}_1 + w_2\vec{x}_2 + \ldots + w_n\vec{x}_n + b$$

```python
def predict(X,w,b):
    """
    X: array (n,k): n training samples, k features
    w: array (k,)
    b: scalar

    return: array (n,)

    """
    k = len(w)
    f = 0
    for i in range(k):
        f += w[i]*X[:,i]     # array (n,) = scalar * array (n,)
    f += b
    return f
```

`predict` for multiple variables (vectorization)

$$f(\vec{x}) = w_1\vec{x}_1 + w_2\vec{x}_2 + \ldots + w_n\vec{x}_n + b = X\vec{w} + b$$

```
def predict(X,w,b):
    """
    X: array (n,k): n training samples, k features
    w: array (k,)
    b: scalar

    return: array (n,)

    """
    # matrix multiplication
    return np.dot(X,w)+b    # array (n,) = array (n,k) * array (k,) + scalar
```

Gradient descent

$$w_j = w_j - \alpha \frac{\partial J(\vec{w},b)}{\partial w_j}, j = 1, 2, \ldots, k$$

$$w_1 = w_1 - \alpha \frac{\partial J(\vec{w},b)}{\partial w_1}$$

$$w_2 = w_2 - \alpha \frac{\partial J(\vec{w},b)}{\partial w_2}$$

…

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

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

Gradient for multivariable regression

$f(\vec{x}) = w_1 x_1 + w_2 x_2 + \ldots + w_k x_k + b$

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

Partial derivative (gradient) of the cost function with respect to w_j

$$\frac{\partial J(\vec{w}, b)}{\partial w_j} = \frac{1}{m} \sum_{i=1}^{m} 2(y^{(i)} - f(\vec{x}^{(i)})) \frac{\partial f(\vec{x})}{\partial w_j} = \frac{1}{m} \sum_{i=1}^{m} 2(y^{(i)} - f(\vec{x}^{(i)}))(-x_j^{(i)})$$

Partial derivative (gradient) of the cost function with respect to b

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

💻 Multivariable regression

Feature scaling

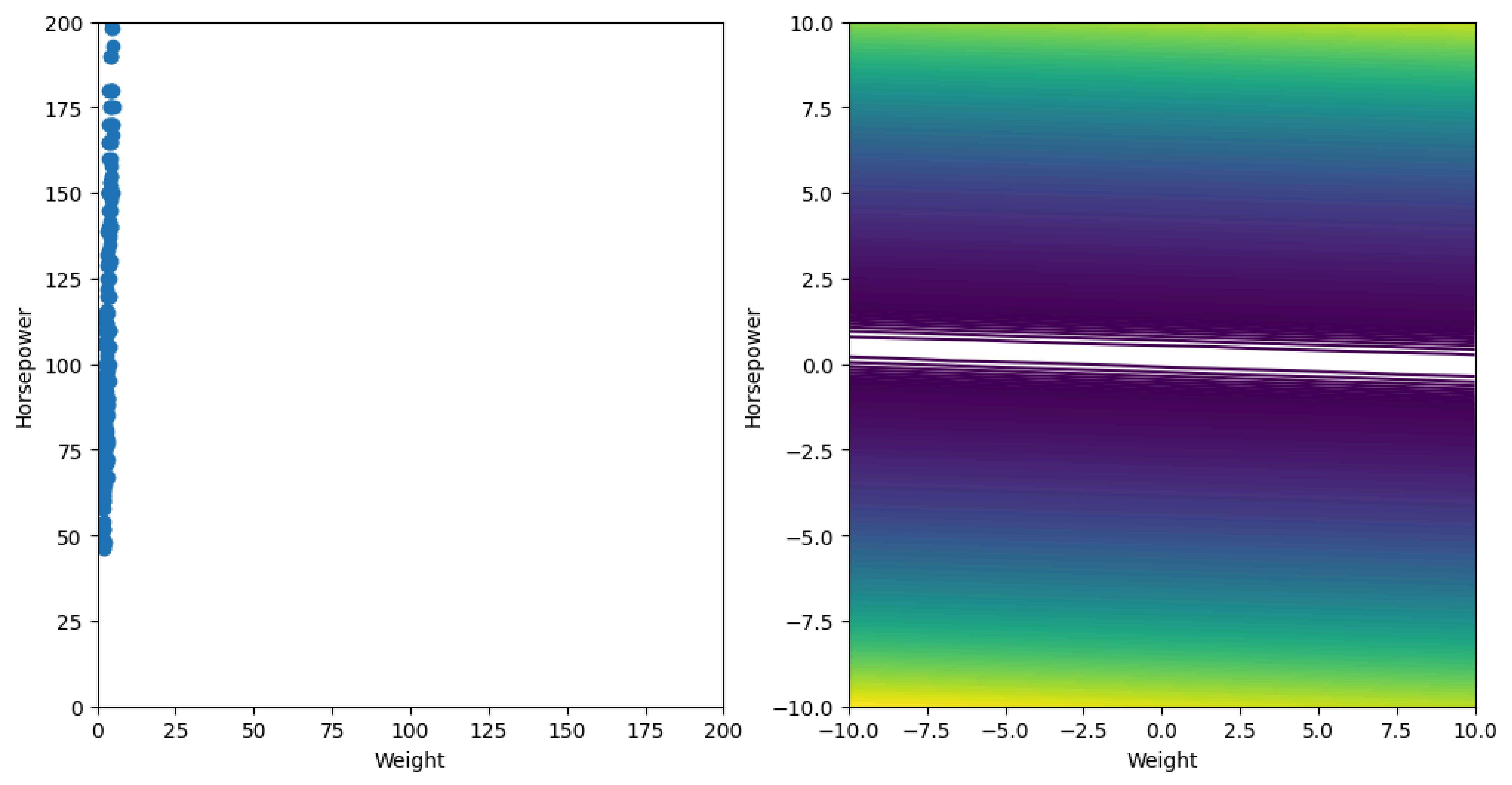
Feature size and parameter size

	size of feature (min-max)	size of parameter
weight	1-5	Large
horsepower	50-300	Small

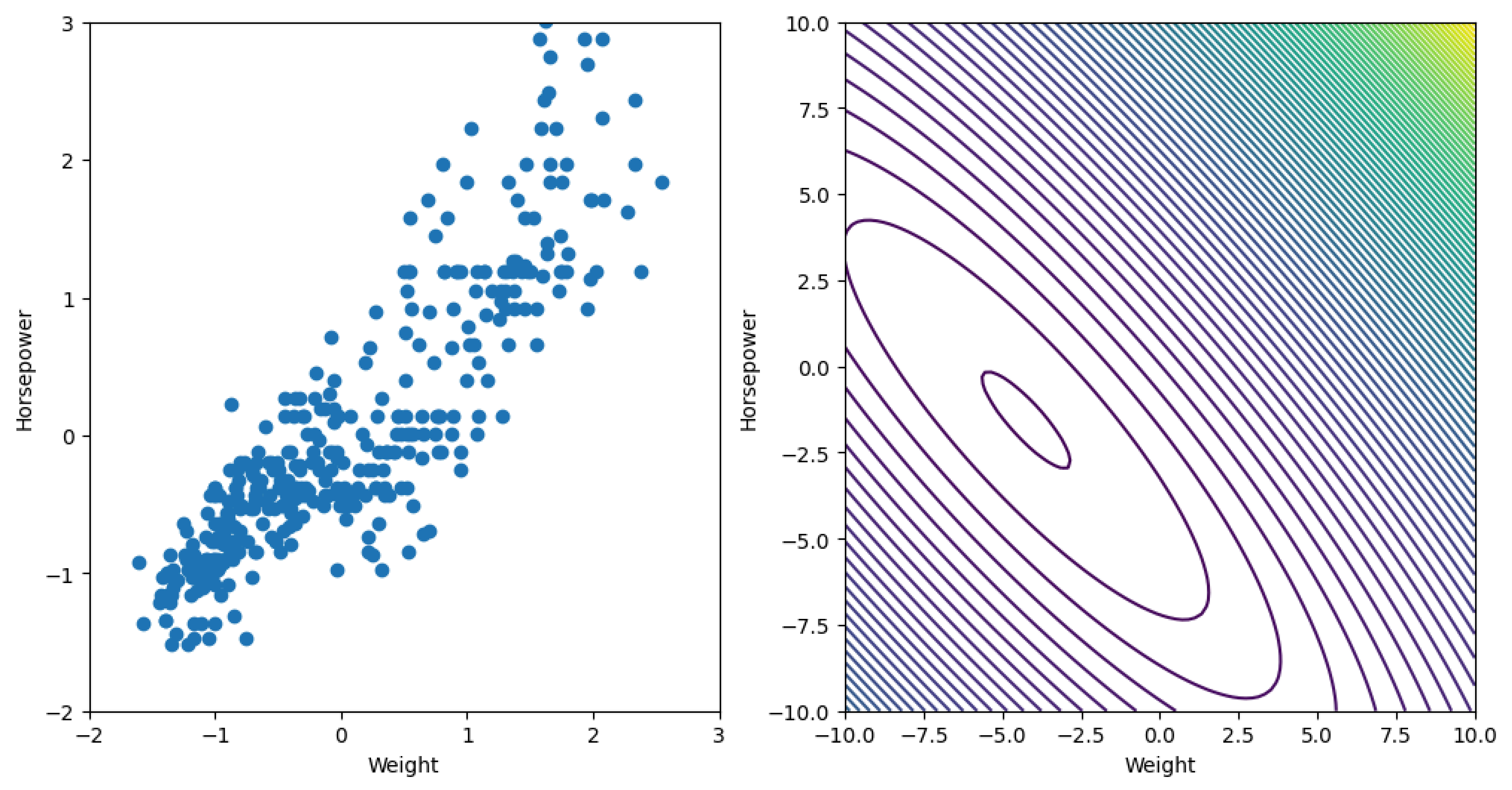
Unbalanced feature size causes:

- narrow cost function
- learning rate needs to be small
- slow convergence or divergence

Feature size and parameter size



Feature size and parameter size



Rescale features

Max normalization

$$x_{scaled} = \frac{x}{x_{max}}$$

Min-max normalization

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Mean normalization

$$x_{scaled} = \frac{x - \mu}{x_{max} - x_{min}}$$

Z-score normalization (standardization)

$$x_{scaled} = \frac{x - \mu}{\sigma}$$

🤔 Rescale features

$$300 \leq x_1 \leq 2000$$
$$0 \leq x_2 \leq 5$$
$$\mu_1 = 600, \mu_2 = 2.3, \sigma_1 = 450, \sigma_2 = 1.4$$

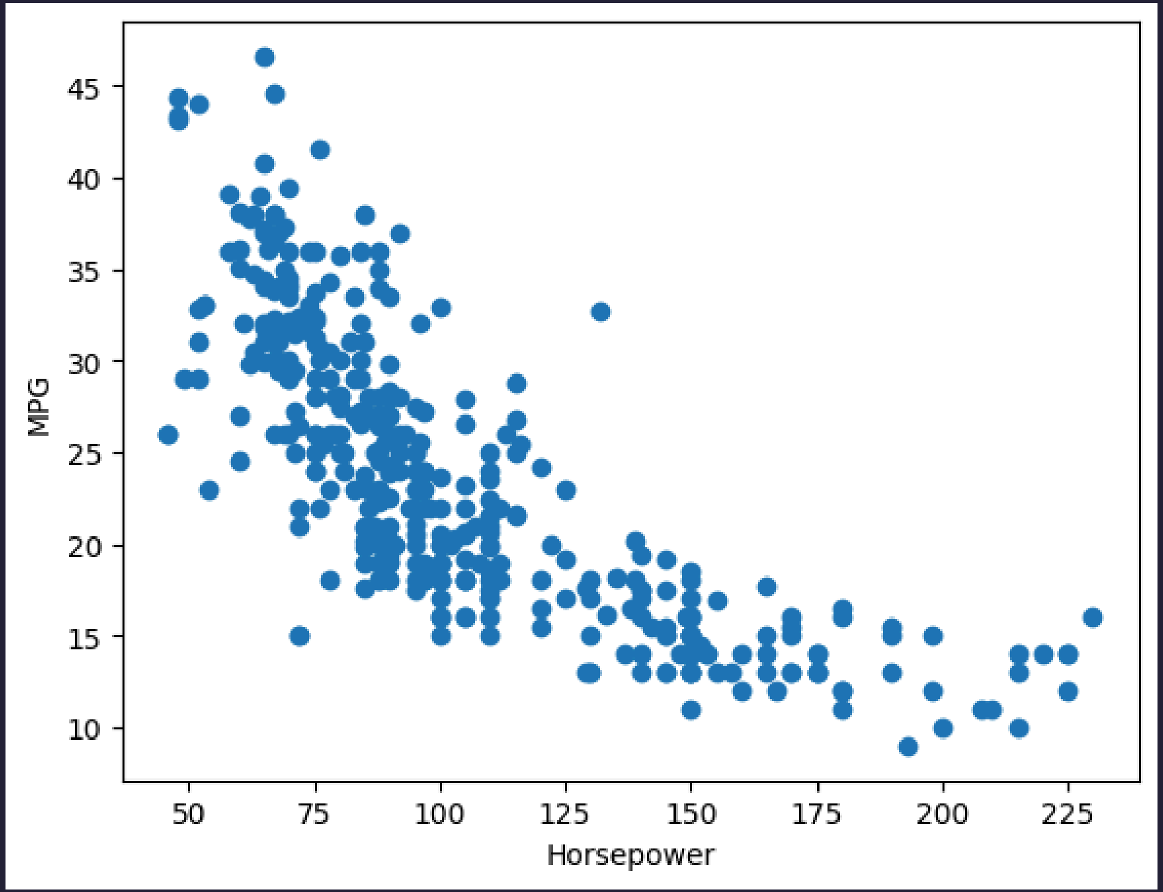
Max normalization?

Min-max normalization?

Mean normalization?

Standardization?

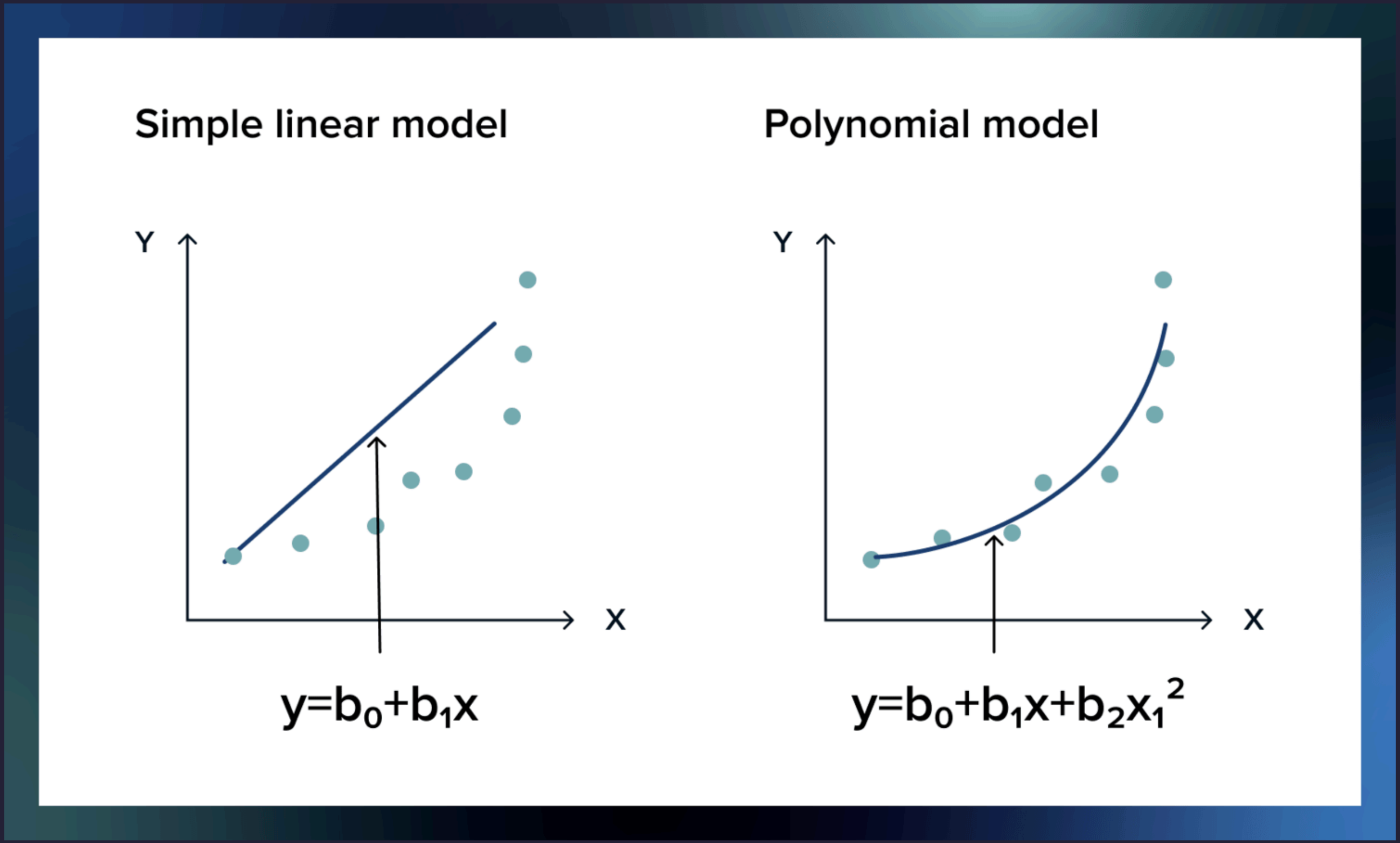
Feature engineering



Numerical features

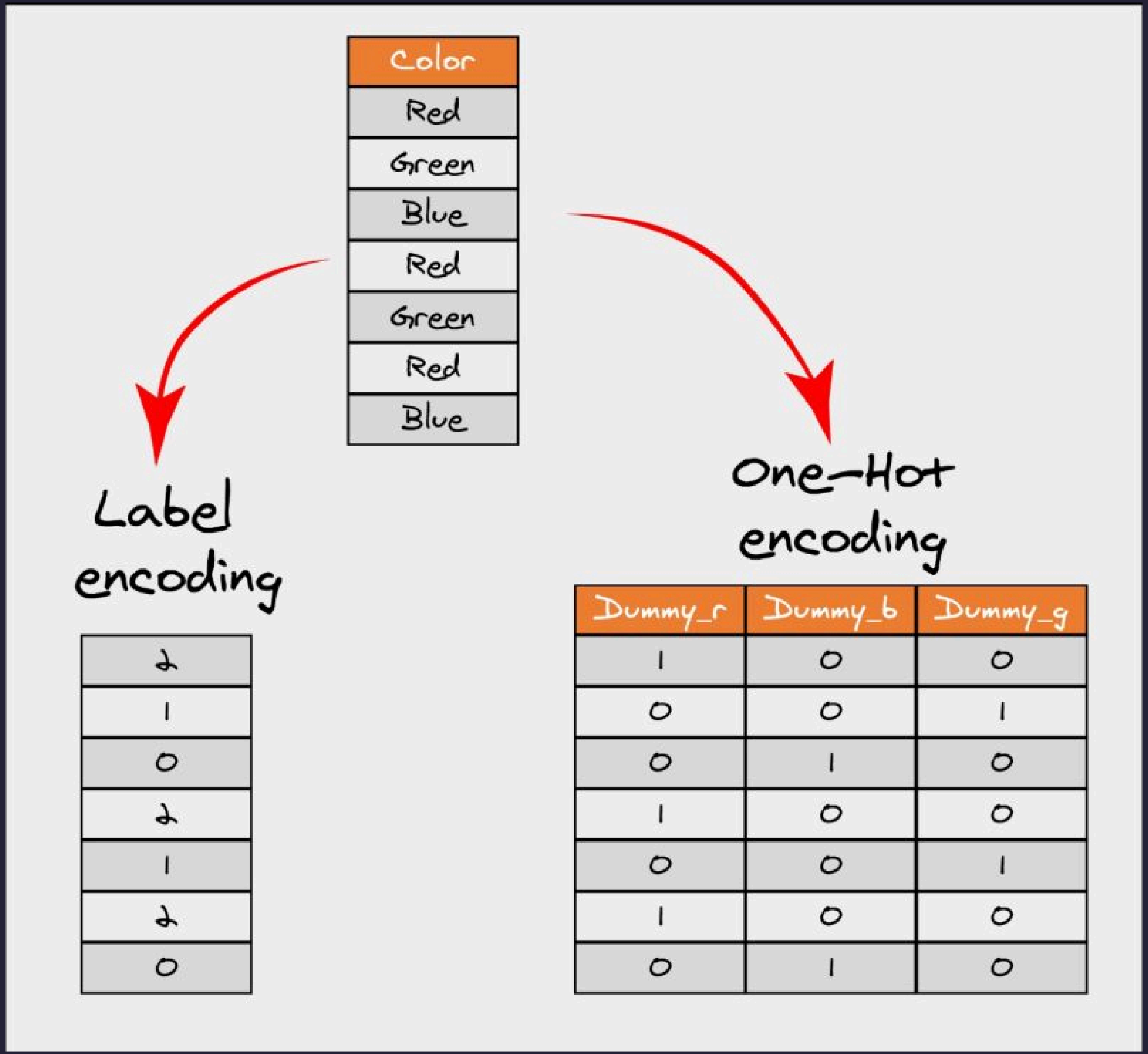
- Polynomial $(x^2, x^3, \sqrt{x}, \ldots)$
  - `x**2,x**3,np.sqrt(x)`
- Interaction $(x_1 x_2)$
  - `x1*x2`
- Logarithmic $(\log x)$
  - `np.log(x)`
- Exponential (e^x)
  - `np.exp(x)`

Polynomial regression



Categorical features

- **Label encoding**: assign a unique integer to each category
- **One-hot encoding** (dummy variables): create binary columns for each category (`pd.get_dummies`)



R-squared

The proportion of the variance in the output that is predictable from the input variable(s)

$$R^2 = 1 - \frac{RSS}{TSS}$$

Total sum of squares:

$$TSS = \sum_{i=1}^{m} (y^{(i)} - \bar{y})^2$$

$$\bar{y} = \frac{1}{m} \sum_{i=1}^{m} y^{(i)}$$

Residual sum of squares:

$$RSS = \sum_{i=1}^{m} (y^{(i)} - f(\vec{x}^{(i)}))^2$$

💻 Feature scaling & engineering