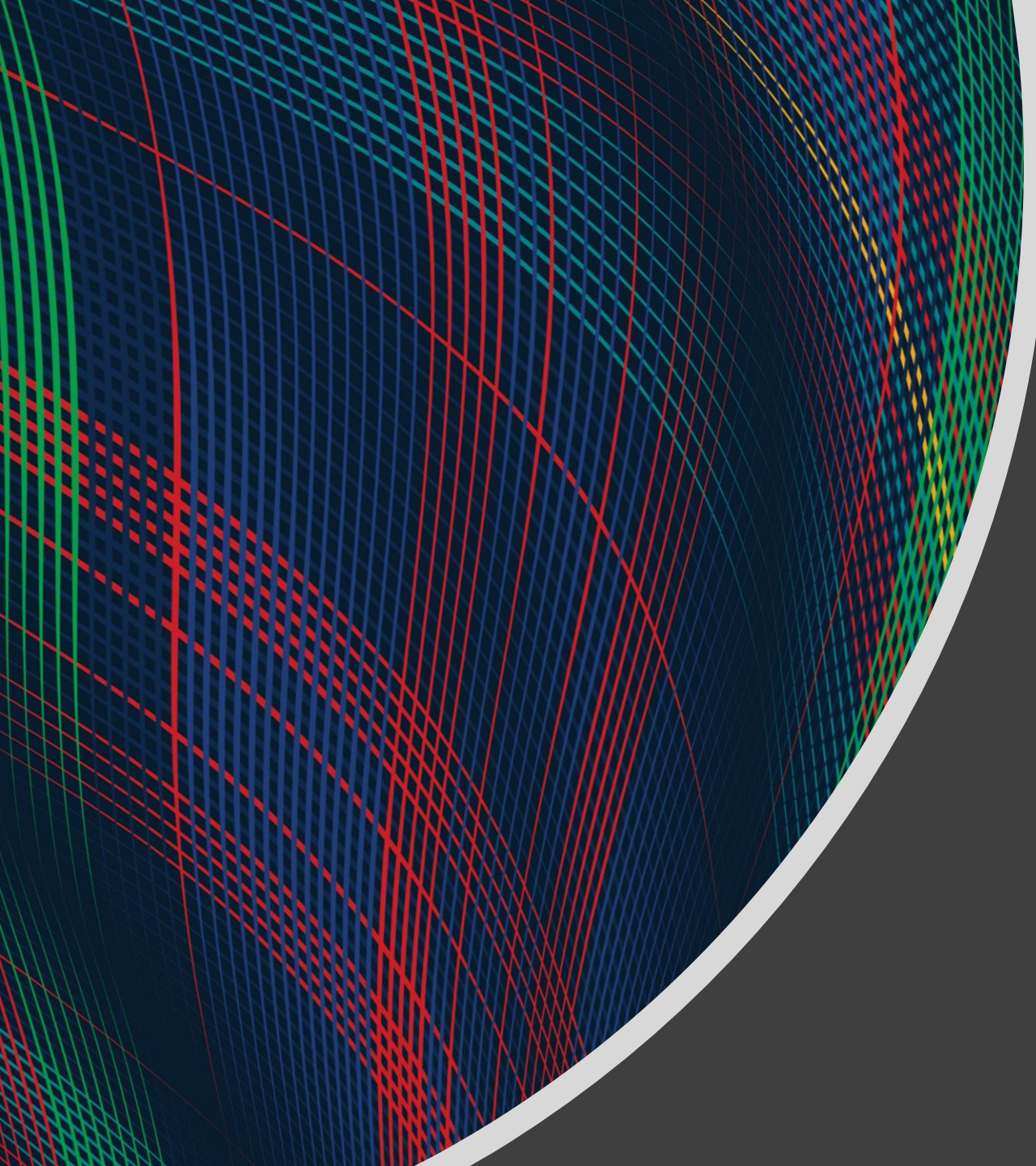


An abstract graphic on the left side of the slide, featuring a sphere-like shape composed of a dense grid of intersecting red, green, and blue lines. The lines are curved and follow the contours of the sphere, creating a complex, woven pattern. The sphere is set against a dark gray background.

10-315

Introduction to ML

Feature Engineering

Instructor: Pat Virtue

# Today

## Student support

## Feature engineering

- Just touch on polynomial feature engineering
- — stop —
- We'll get back to more feature engineering and feature learning later in the semester

## Logistic Regression



## Poll 1

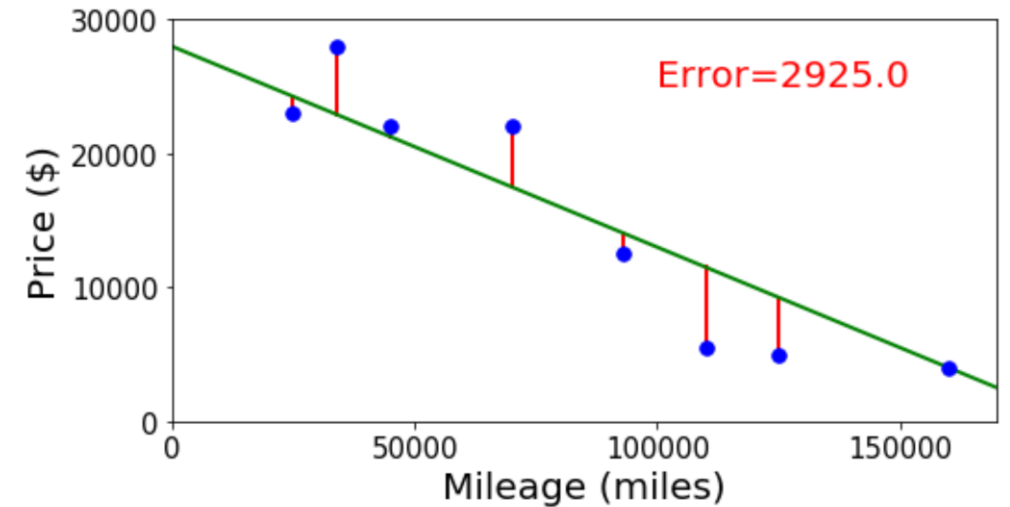
How many appointment slots does Pat have available this Friday?

## Poll 2

Linear regression. What is linear about it?

Select all that apply.

- A. Always fits a linear (or affine) shape to the data
- B. Linear objective function with respect to the input
- C. Linear objective function with respect to the parameters
- D. None of the above

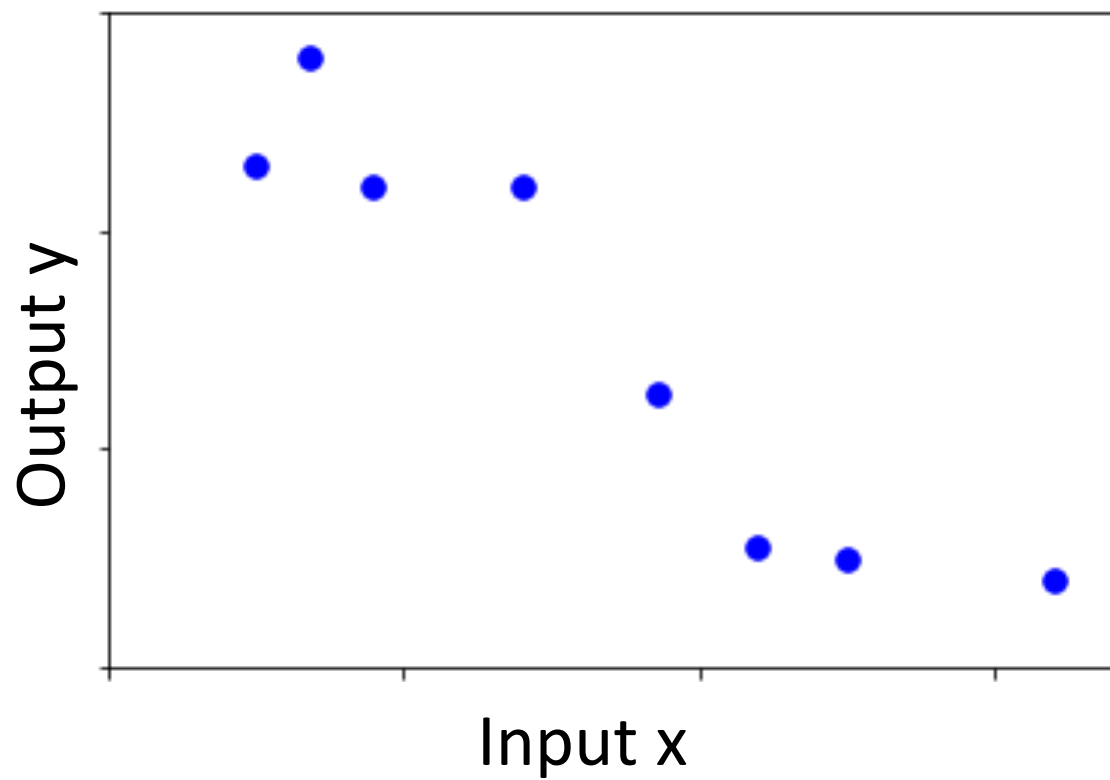


Pre-reading

Polynomial Features

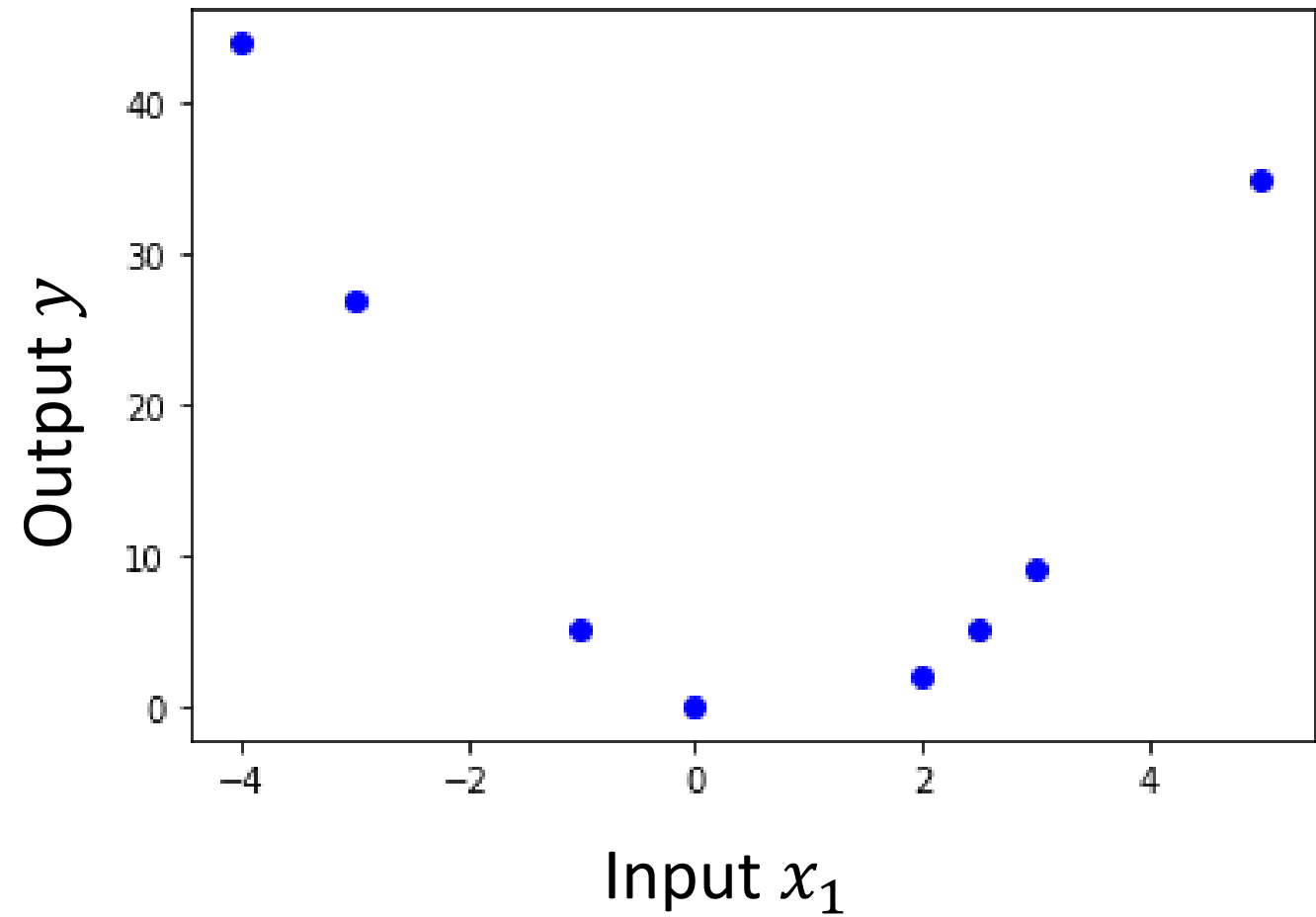
# Linear Data

Regression on simple linear dataset



# Non-linear Data

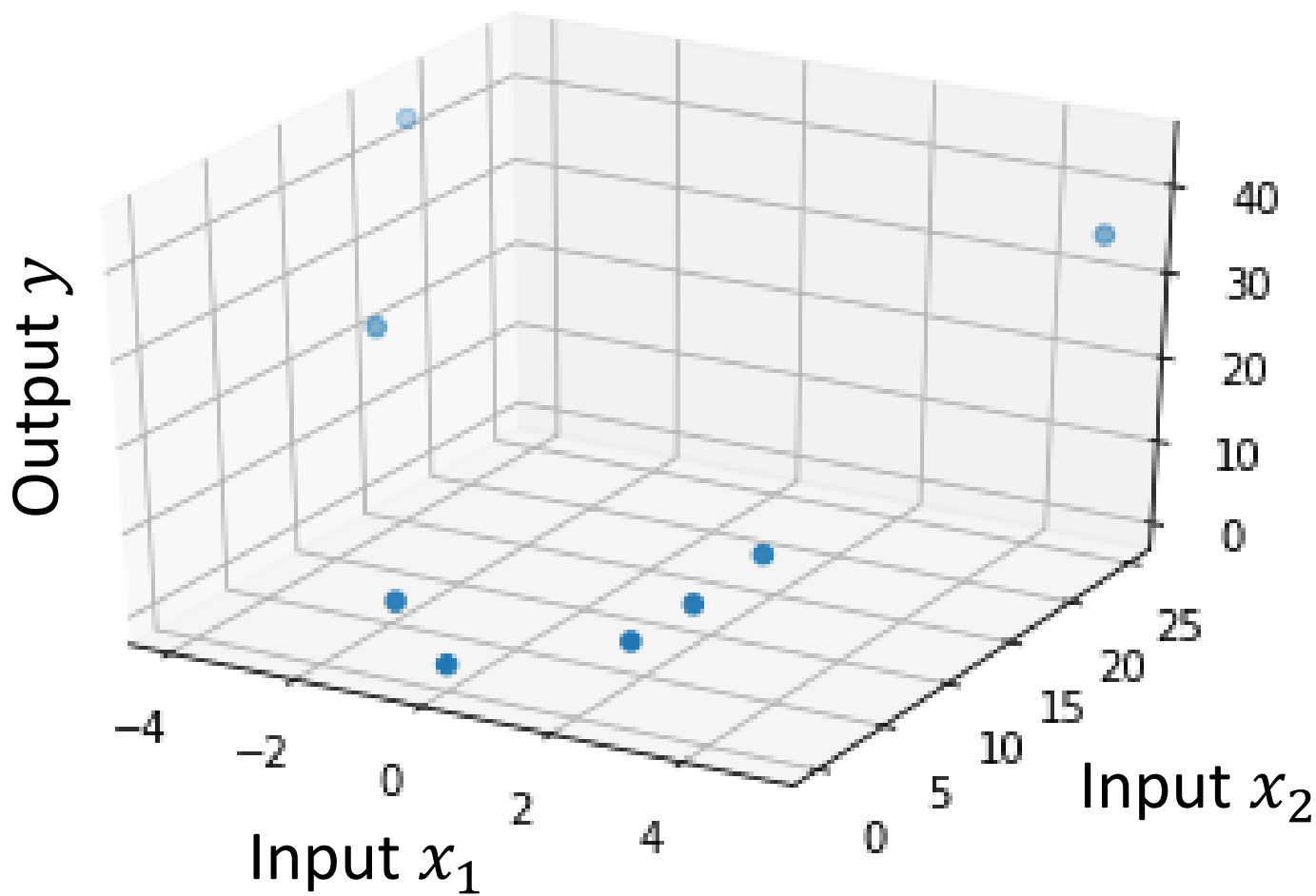
Linear regression on polynomial data?



| $y$ | $x_1$ |
|-----|-------|
| 44  | -4    |
| 27  | -3    |
| 5   | -1    |
| 0   | 0     |
| 2   | 2     |
| 5   | 2.5   |
| 9   | 3     |
| 35  | 5     |

# Non-linear Data

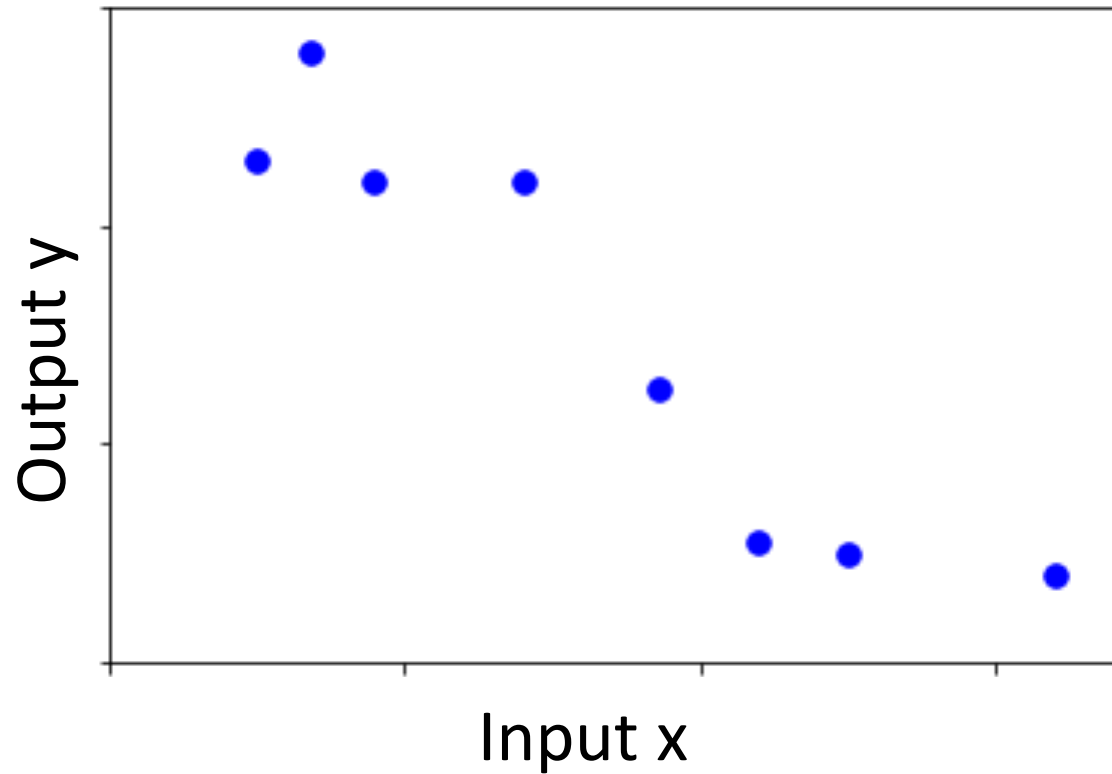
Regression on simple linear dataset



| $y$ | $x_1$ | $x_2$ |
|-----|-------|-------|
| 44  | -4    | 16    |
| 27  | -3    | 9     |
| 5   | -1    | 1     |
| 0   | 0     | 0     |
| 2   | 2     | 4     |
| 5   | 2.5   | 6.25  |
| 9   | 3     | 9     |
| 35  | 5     | 25    |

# Non-linear Data

Polynomial feature map for linear regression

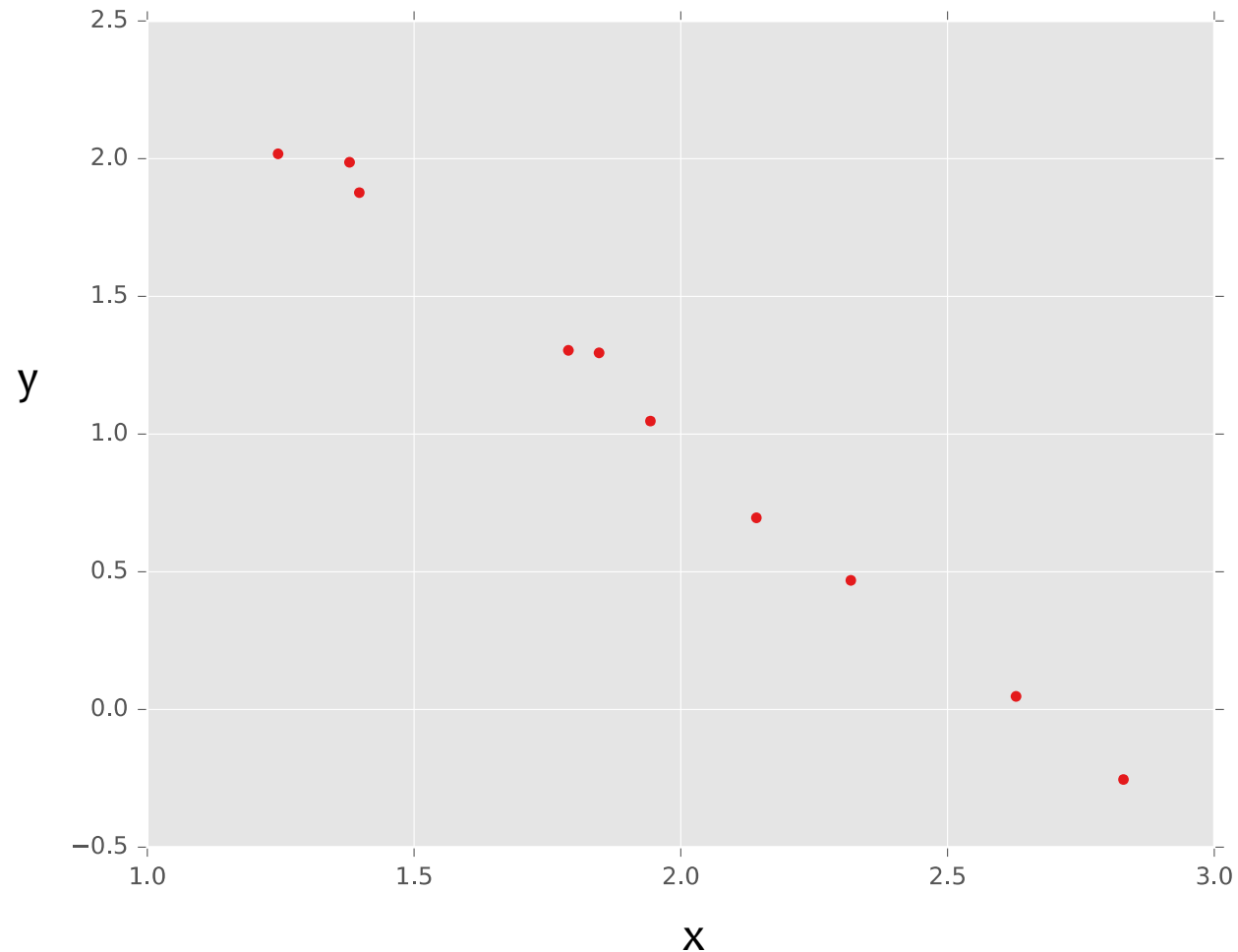


# Example: Linear Regression

**Goal:** Learn  $y = \mathbf{w}^T \mathbf{f}(\mathbf{x}) + b$   
where  $\mathbf{f}(\cdot)$  is a polynomial  
basis function

| y   | x   |
|-----|-----|
| 2.0 | 1.2 |
| 1.3 | 1.7 |
| 0.1 | 2.7 |
| 1.1 | 1.9 |

true “unknown”  
target function is  
linear with  
negative slope  
and gaussian  
noise

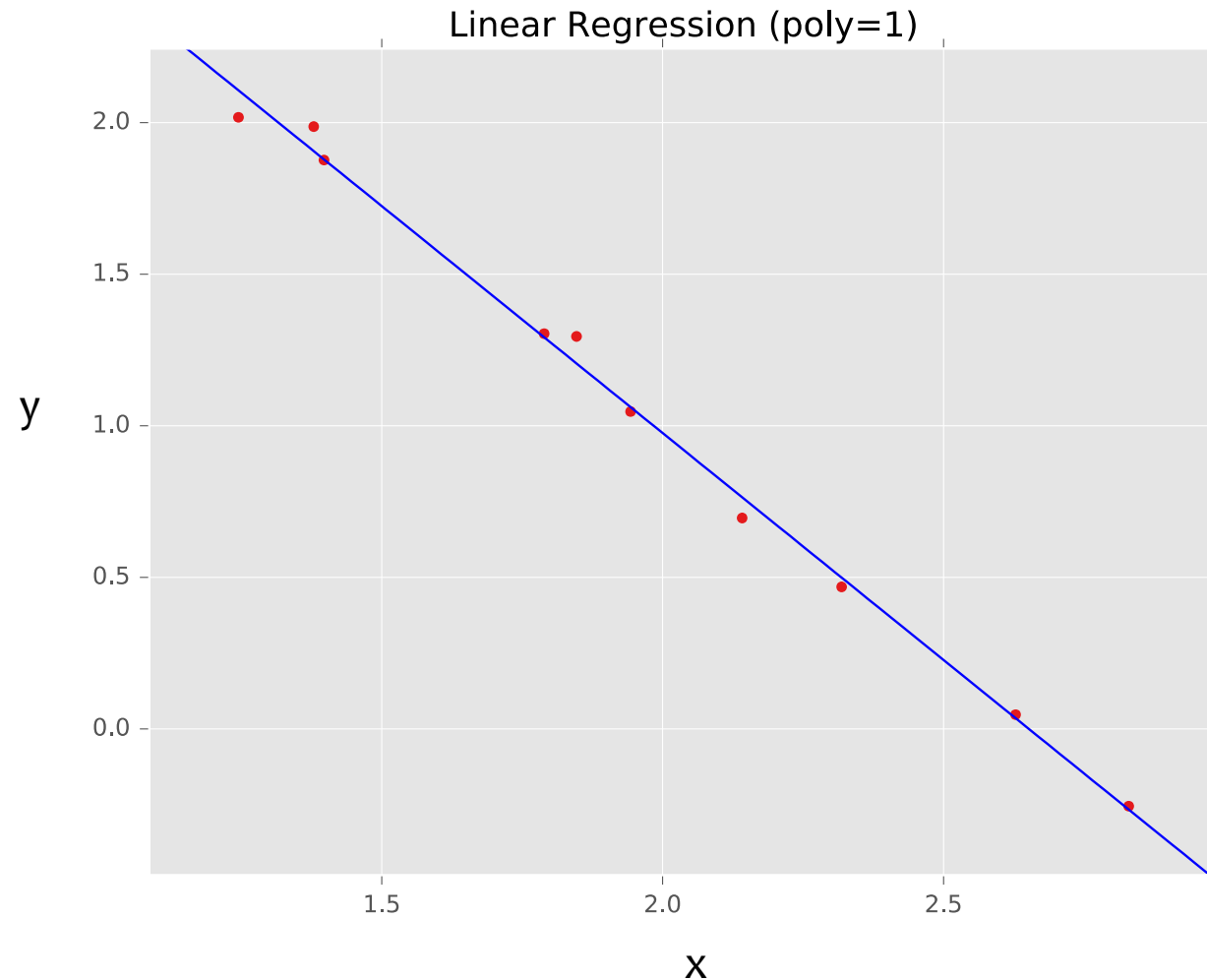


# Example: Linear Regression

**Goal:** Learn  $y = \mathbf{w}^T \mathbf{f}(\mathbf{x}) + b$   
where  $\mathbf{f}(\cdot)$  is a polynomial  
basis function

| y   | x   |
|-----|-----|
| 2.0 | 1.2 |
| 1.3 | 1.7 |
| 0.1 | 2.7 |
| 1.1 | 1.9 |

true “unknown”  
target function is  
linear with  
negative slope  
and gaussian  
noise

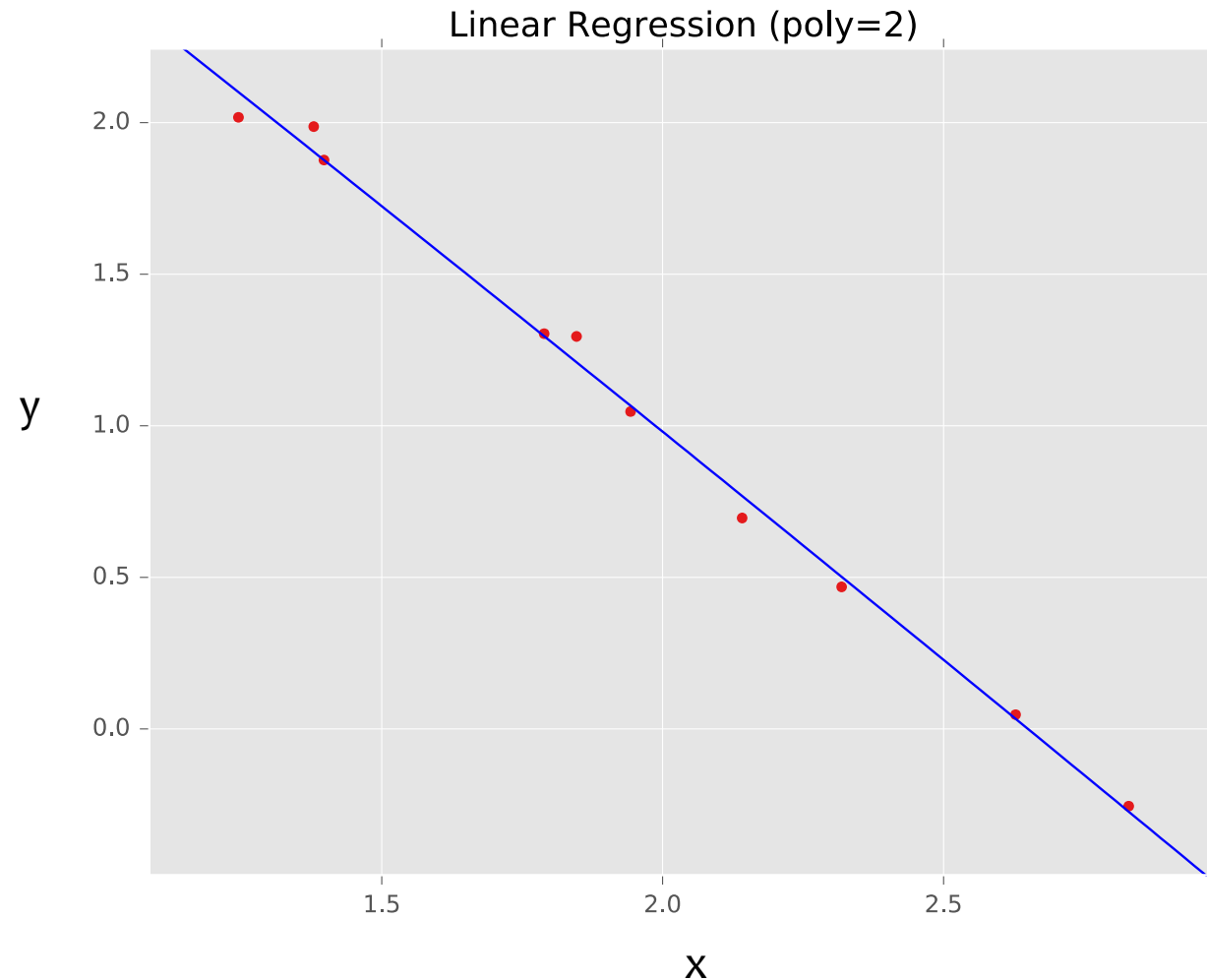


# Example: Linear Regression

**Goal:** Learn  $y = \mathbf{w}^T \mathbf{f}(\mathbf{x}) + b$   
where  $\mathbf{f}(\cdot)$  is a polynomial  
basis function

| y   | x   | $x^2$     |
|-----|-----|-----------|
| 2.0 | 1.2 | $(1.2)^2$ |
| 1.3 | 1.7 | $(1.7)^2$ |
| 0.1 | 2.7 | $(2.7)^2$ |
| 1.1 | 1.9 | $(1.9)^2$ |

true “unknown”  
target function is  
linear with  
negative slope  
and gaussian  
noise

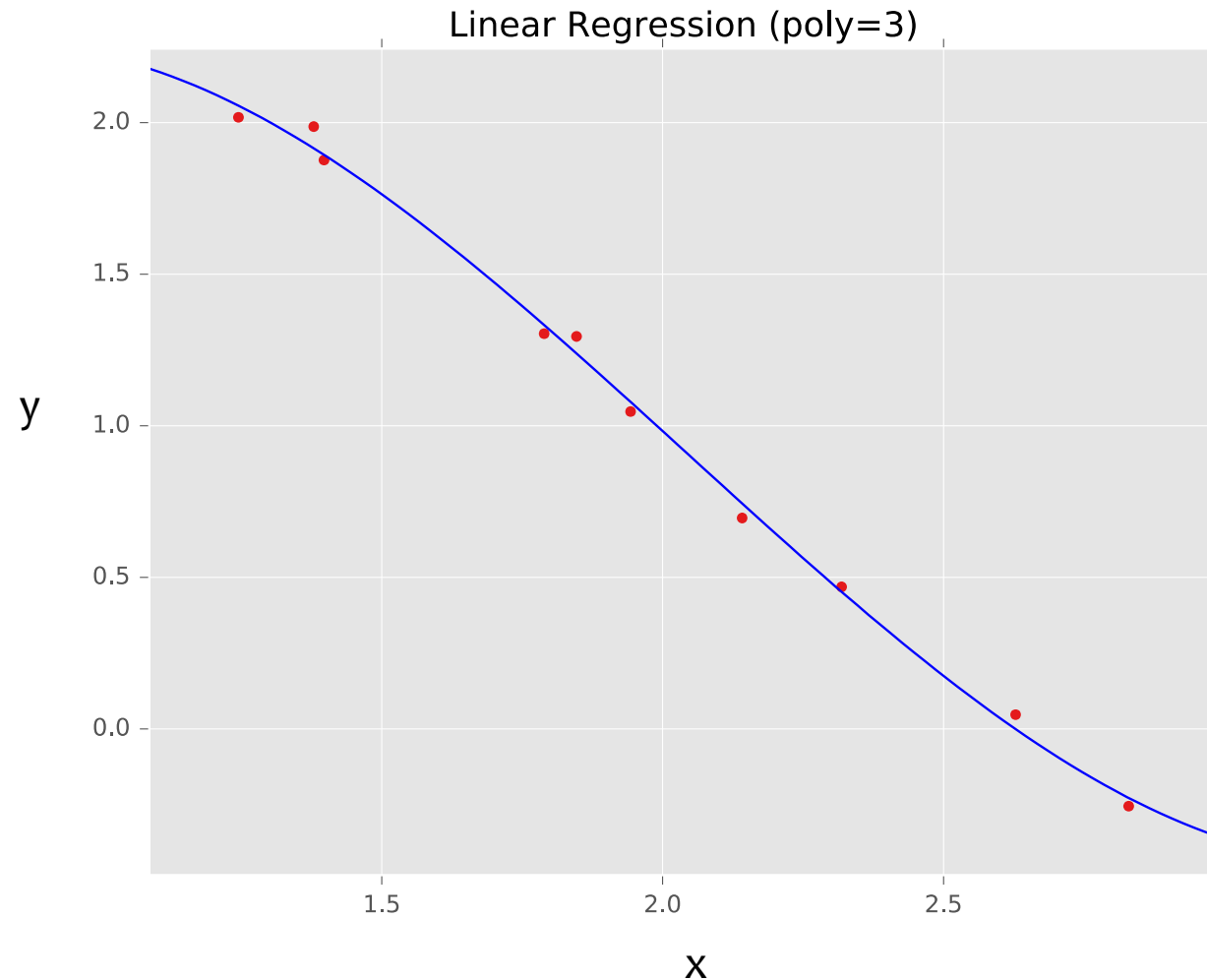


# Example: Linear Regression

**Goal:** Learn  $y = \mathbf{w}^T \mathbf{f}(\mathbf{x}) + b$   
where  $\mathbf{f}(\cdot)$  is a polynomial  
basis function

| y   | x   | $x^2$     | $x^3$     |
|-----|-----|-----------|-----------|
| 2.0 | 1.2 | $(1.2)^2$ | $(1.2)^3$ |
| 1.3 | 1.7 | $(1.7)^2$ | $(1.7)^3$ |
| 0.1 | 2.7 | $(2.7)^2$ | $(2.7)^3$ |
| 1.1 | 1.9 | $(1.9)^2$ | $(1.9)^3$ |

true “unknown”  
target function is  
linear with  
negative slope  
and gaussian  
noise

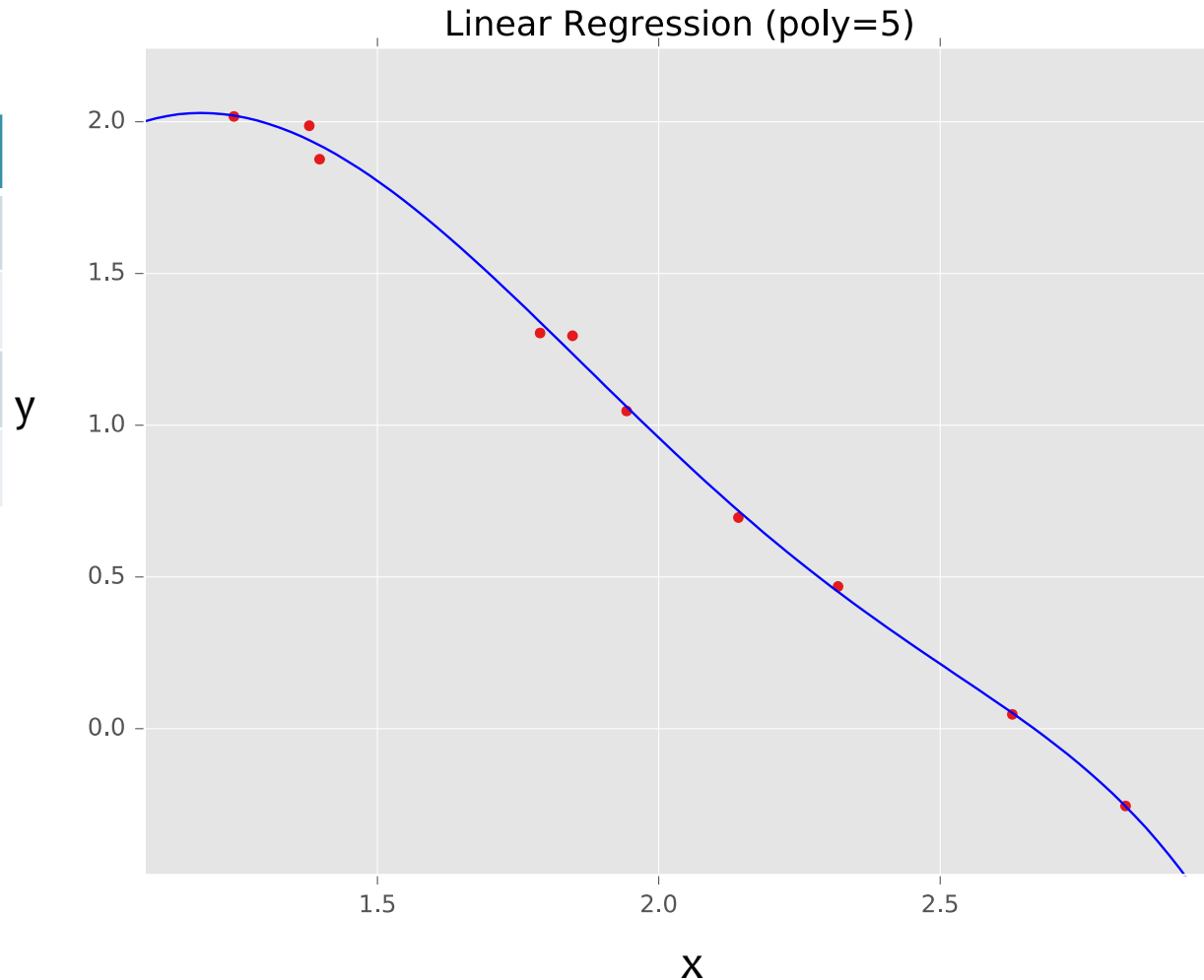


# Example: Linear Regression

**Goal:** Learn  $y = \mathbf{w}^T \mathbf{f}(\mathbf{x}) + b$   
where  $\mathbf{f}(\cdot)$  is a polynomial  
basis function

| y   | x   | $x^2$     | ... | $x^5$     |
|-----|-----|-----------|-----|-----------|
| 2.0 | 1.2 | $(1.2)^2$ | ... | $(1.2)^5$ |
| 1.3 | 1.7 | $(1.7)^2$ | ... | $(1.7)^5$ |
| 0.1 | 2.7 | $(2.7)^2$ | ... | $(2.7)^5$ |
| 1.1 | 1.9 | $(1.9)^2$ | ... | $(1.9)^5$ |

true “unknown”  
target function is  
linear with  
negative slope  
and gaussian  
noise

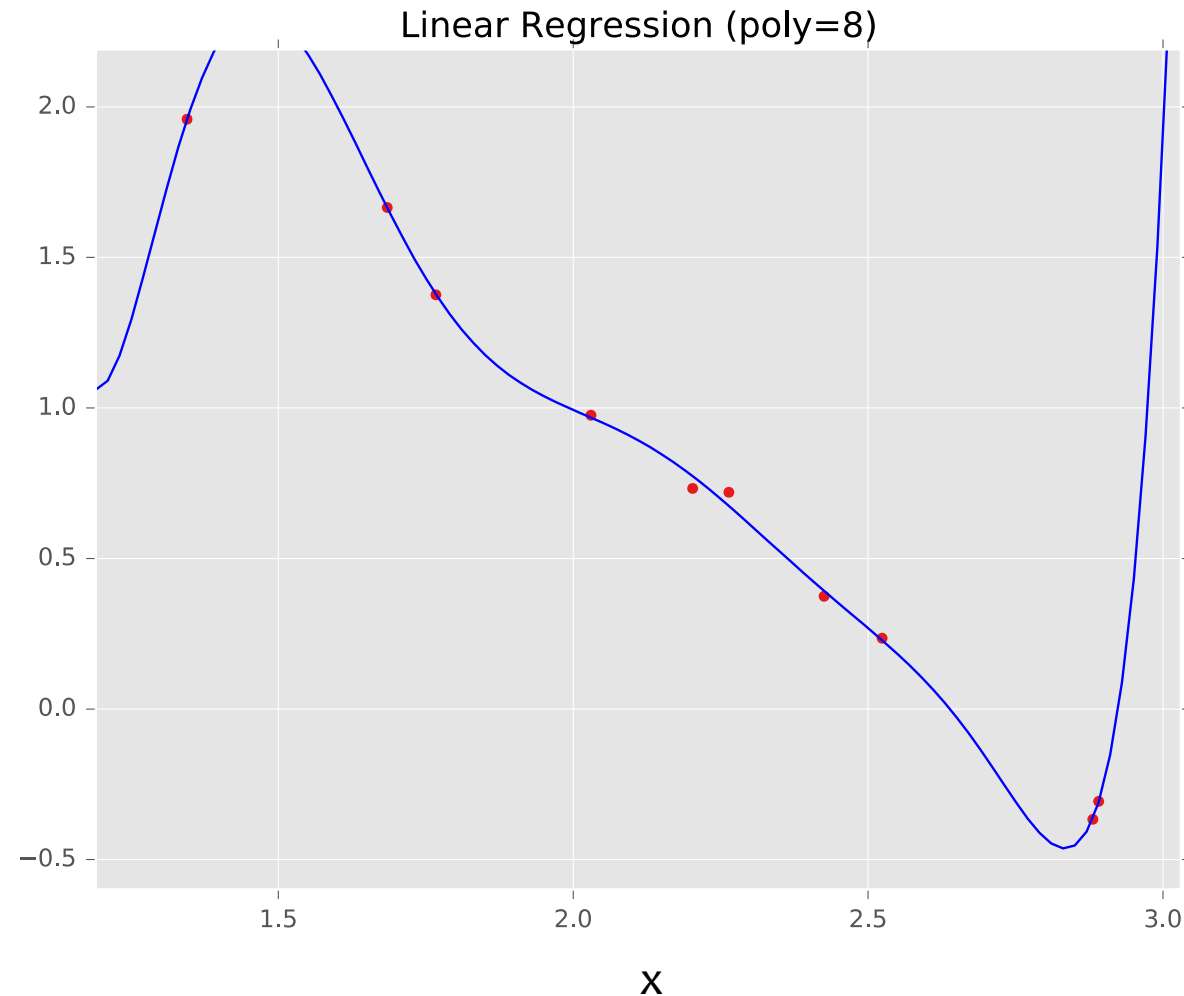


# Example: Linear Regression

**Goal:** Learn  $y = \mathbf{w}^T \mathbf{f}(\mathbf{x}) + b$   
where  $\mathbf{f}(\cdot)$  is a polynomial  
basis function

| y   | x   | $x^2$     | ... | $x^8$     |
|-----|-----|-----------|-----|-----------|
| 2.0 | 1.2 | $(1.2)^2$ | ... | $(1.2)^8$ |
| 1.3 | 1.7 | $(1.7)^2$ | ... | $(1.7)^8$ |
| 0.1 | 2.7 | $(2.7)^2$ | ... | $(2.7)^8$ |
| 1.1 | 1.9 | $(1.9)^2$ | ... | $(1.9)^8$ |

true “unknown”  
target function is  
linear with  
negative slope  
and gaussian  
noise

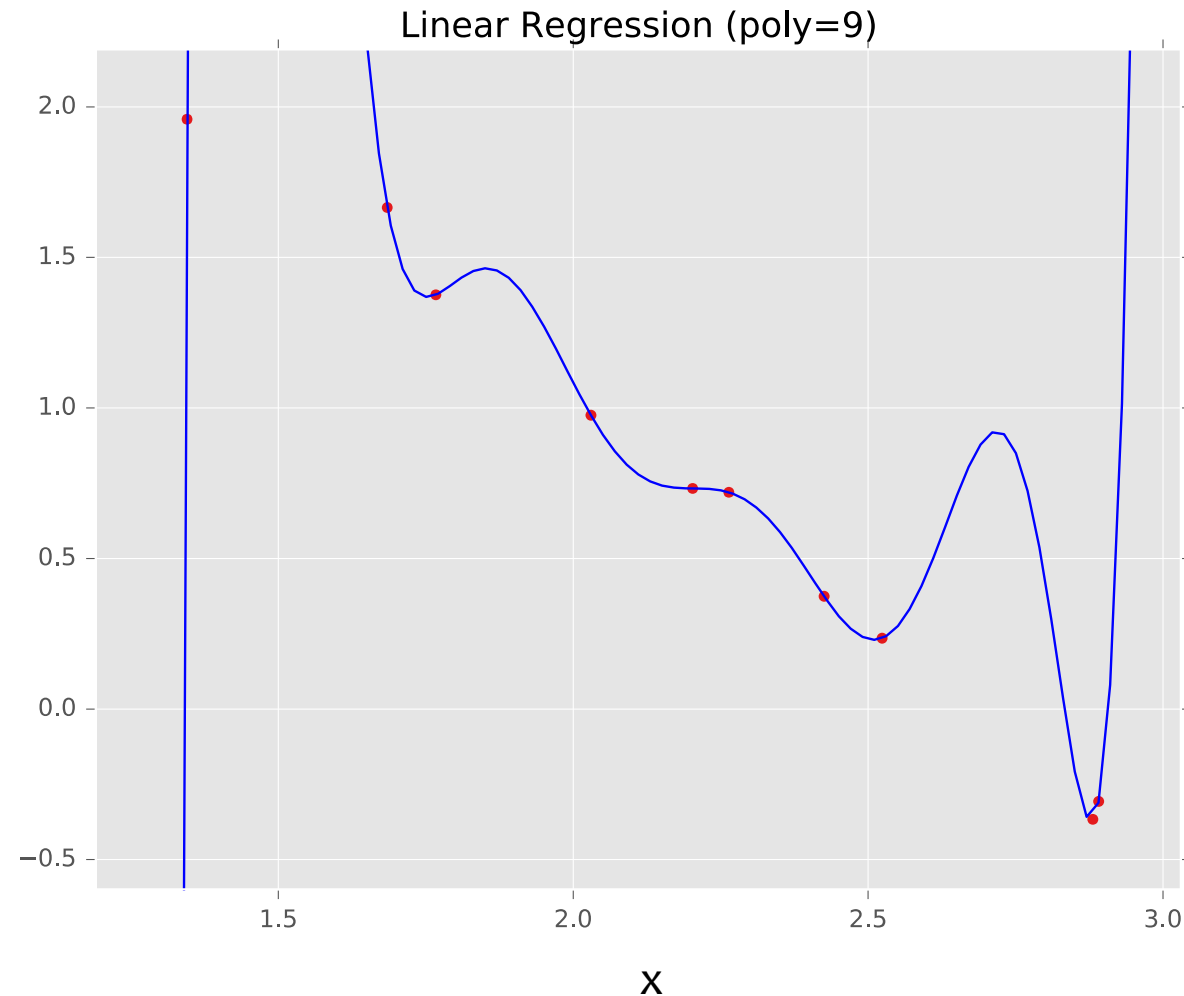


# Example: Linear Regression

**Goal:** Learn  $y = \mathbf{w}^T \mathbf{f}(\mathbf{x}) + b$   
where  $\mathbf{f}(\cdot)$  is a polynomial  
basis function

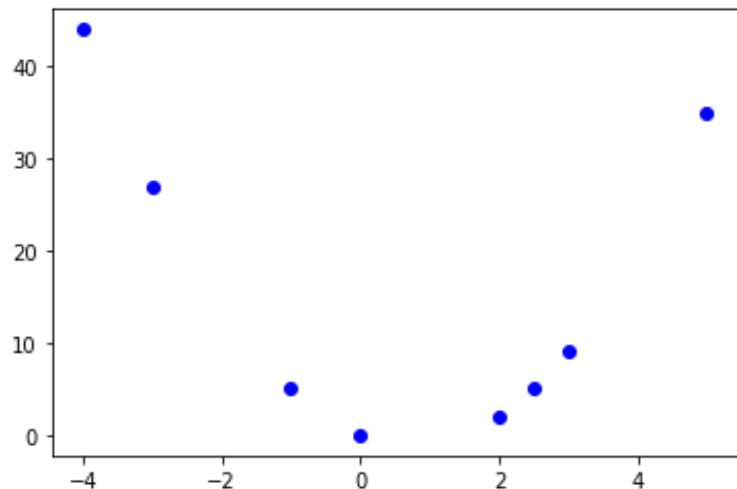
| y   | x   | $x^2$     | ... | $x^9$     |
|-----|-----|-----------|-----|-----------|
| 2.0 | 1.2 | $(1.2)^2$ | ... | $(1.2)^9$ |
| 1.3 | 1.7 | $(1.7)^2$ | ... | $(1.7)^9$ |
| 0.1 | 2.7 | $(2.7)^2$ | ... | $(2.7)^9$ |
| 1.1 | 1.9 | $(1.9)^2$ | ... | $(1.9)^9$ |

true “unknown”  
target function is  
linear with  
negative slope  
and gaussian  
noise



# Polynomial Features

## Design Matrix



| $y$ | $x_1$ |
|-----|-------|
| 44  | -4    |
| 27  | -3    |
| 5   | -1    |
| 0   | 0     |
| 2   | 2     |
| 5   | 2.5   |
| 9   | 3     |
| 35  | 5     |

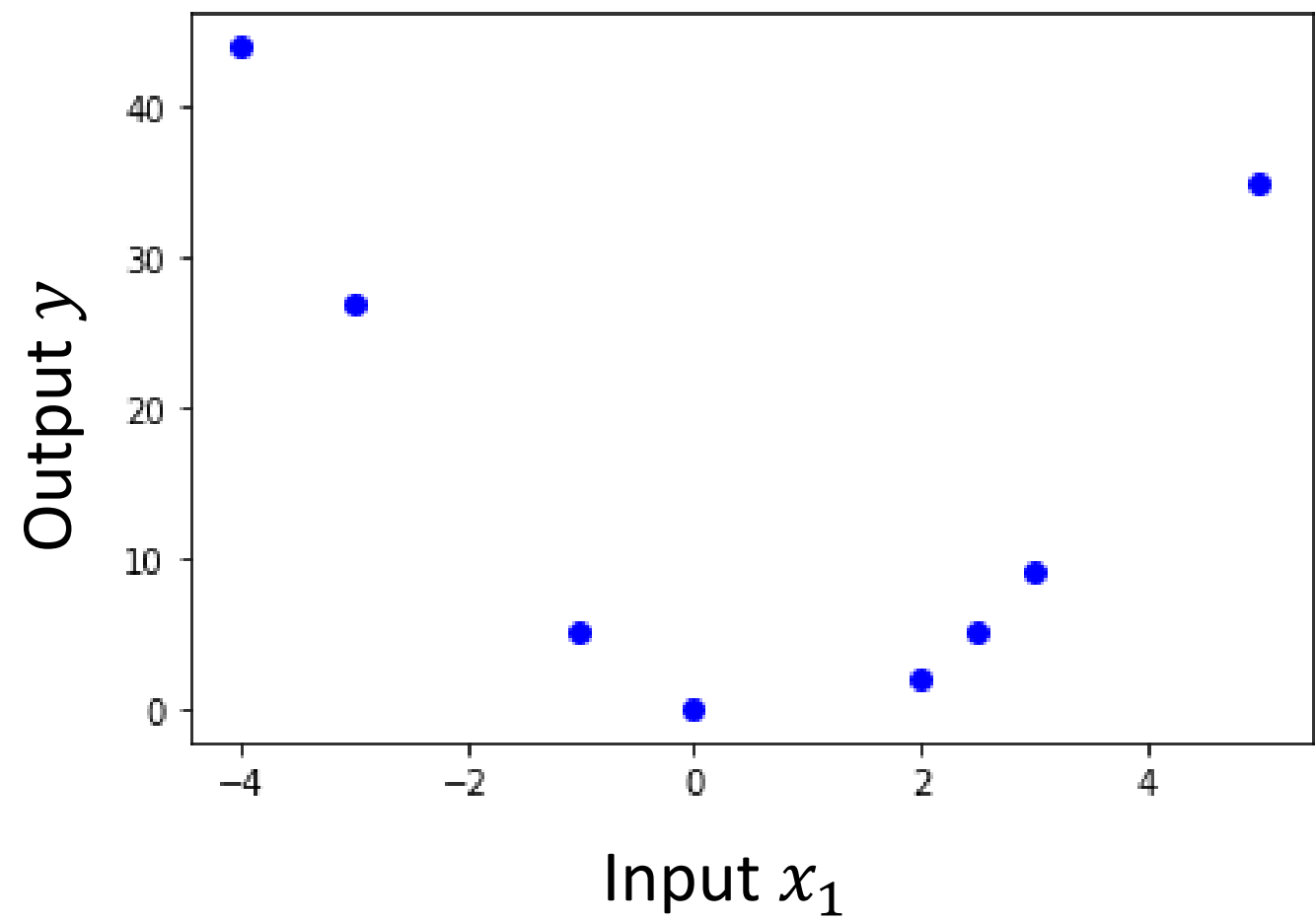
$$X = \begin{bmatrix} \text{---} \mathbf{x}^{(1)\top} \text{---} \\ \text{---} \mathbf{x}^{(2)\top} \text{---} \\ \text{---} \mathbf{x}^{(3)\top} \text{---} \\ \vdots \\ \text{---} \mathbf{x}^{(N)\top} \text{---} \end{bmatrix}$$

$$\phi(X) = \begin{bmatrix} \text{---} \phi(\mathbf{x}^{(1)})^\top \text{---} \\ \text{---} \phi(\mathbf{x}^{(2)})^\top \text{---} \\ \text{---} \phi(\mathbf{x}^{(3)})^\top \text{---} \\ \vdots \\ \text{---} \phi(\mathbf{x}^{(N)})^\top \text{---} \end{bmatrix}$$

# Polynomial Features

# Non-linear Data

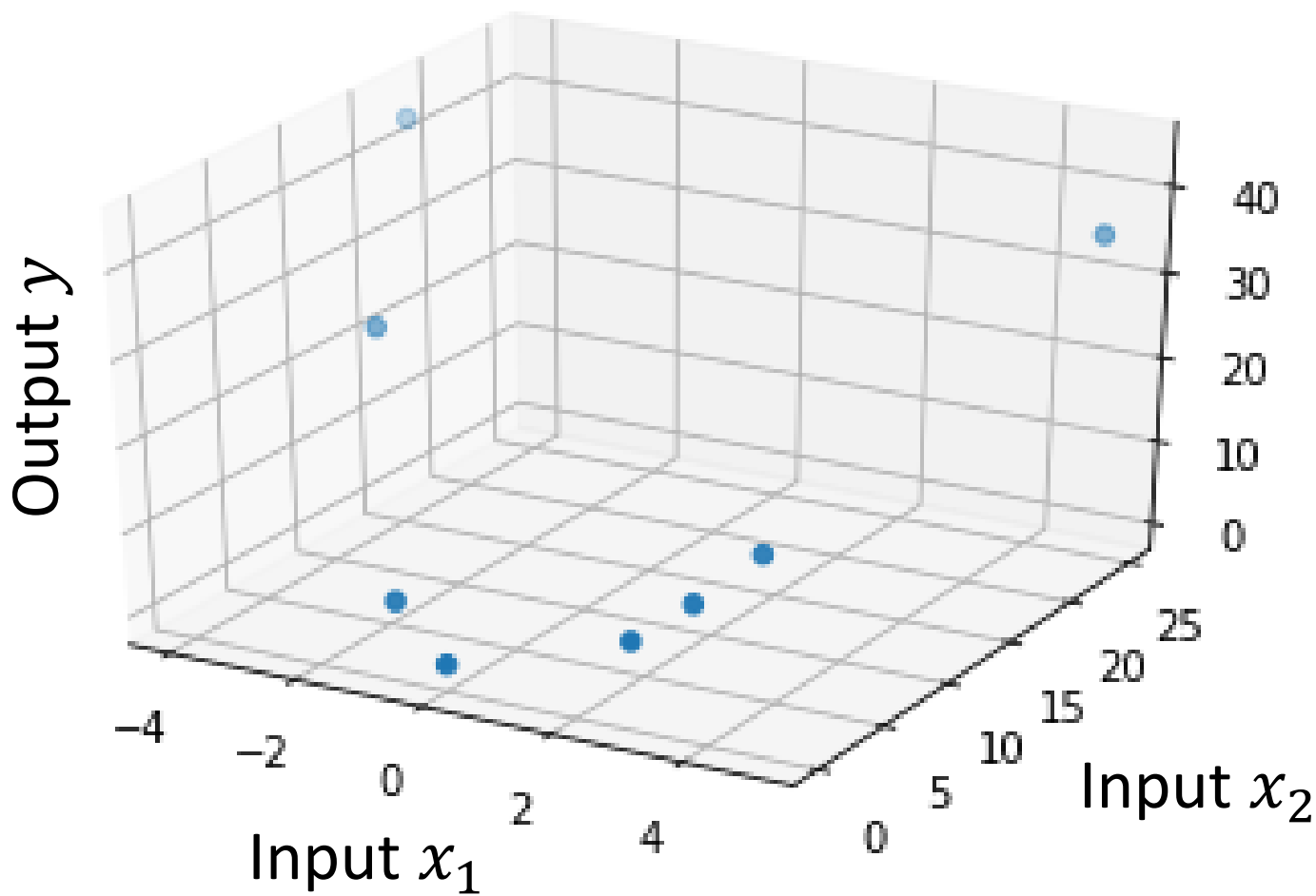
Linear regression on polynomial data?



| $y$ | $x_1$ |
|-----|-------|
| 44  | -4    |
| 27  | -3    |
| 5   | -1    |
| 0   | 0     |
| 2   | 2     |
| 5   | 2.5   |
| 9   | 3     |
| 35  | 5     |

# Non-linear Data

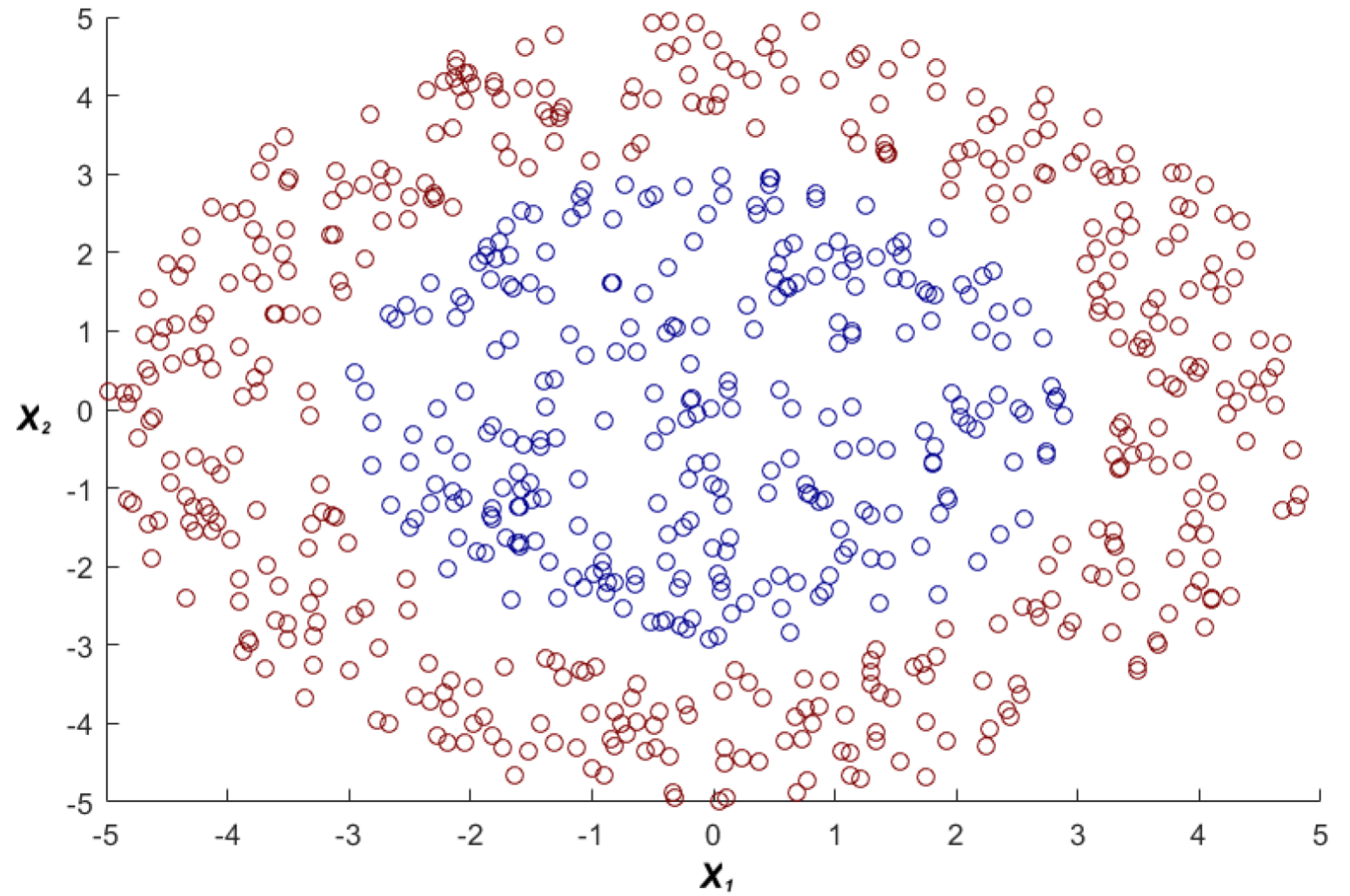
Regression on simple linear dataset



| $y$ | $x_1$ | $x_2$ |
|-----|-------|-------|
| 44  | -4    | 16    |
| 27  | -3    | 9     |
| 5   | -1    | 1     |
| 0   | 0     | 0     |
| 2   | 2     | 4     |
| 5   | 2.5   | 6.25  |
| 9   | 3     | 9     |
| 35  | 5     | 25    |

# Non-linear Data

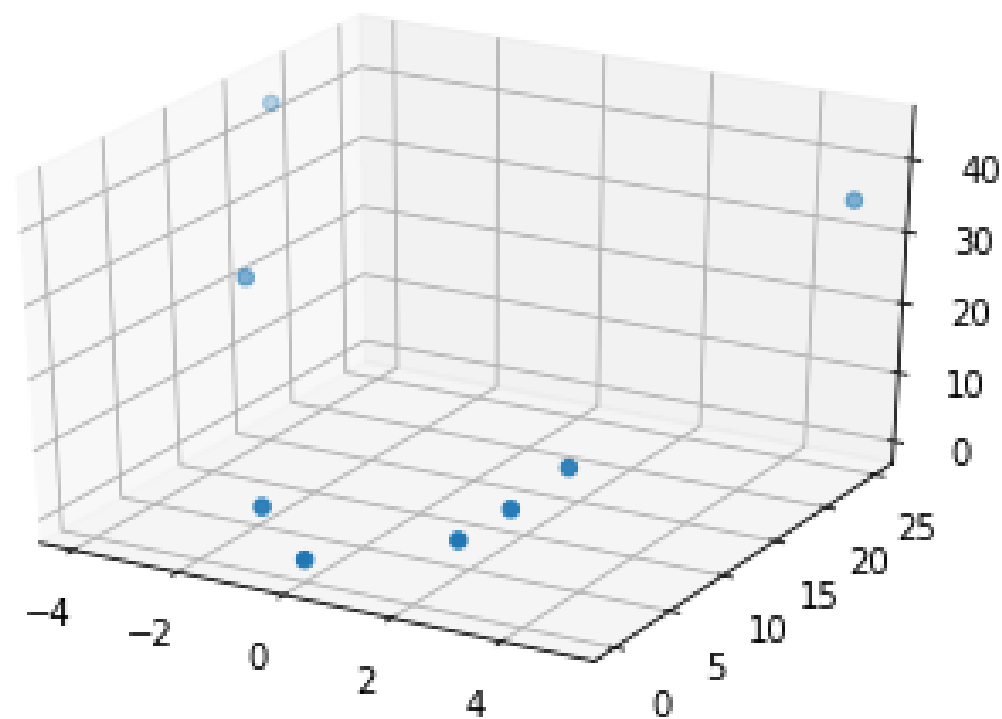
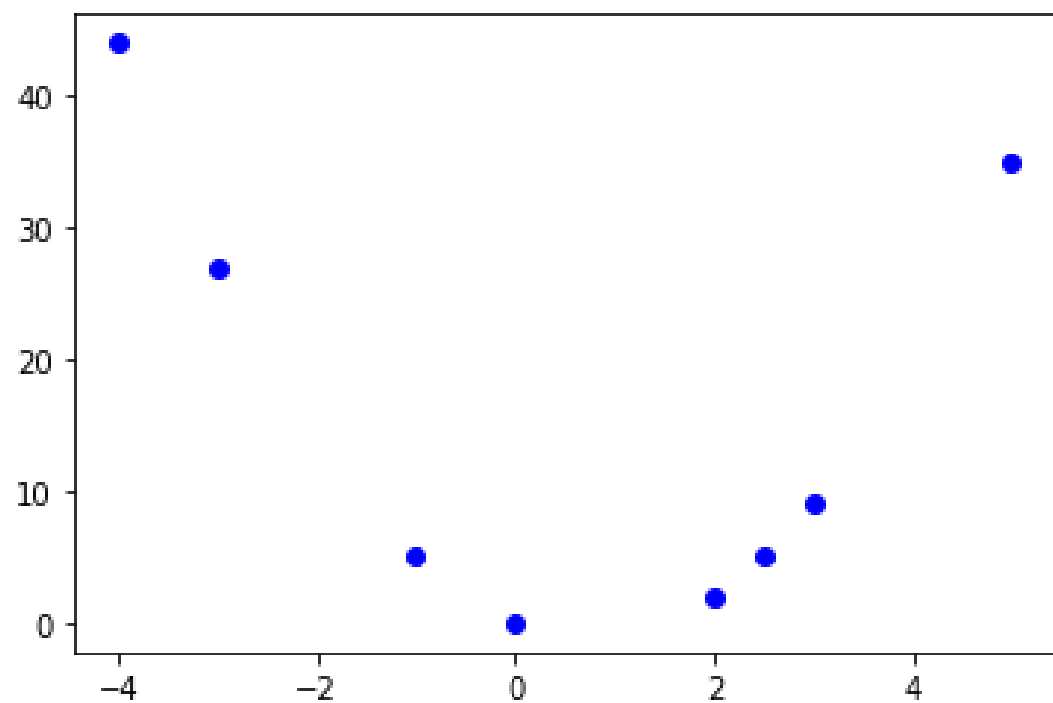
Polynomial feature map  
for linear classification



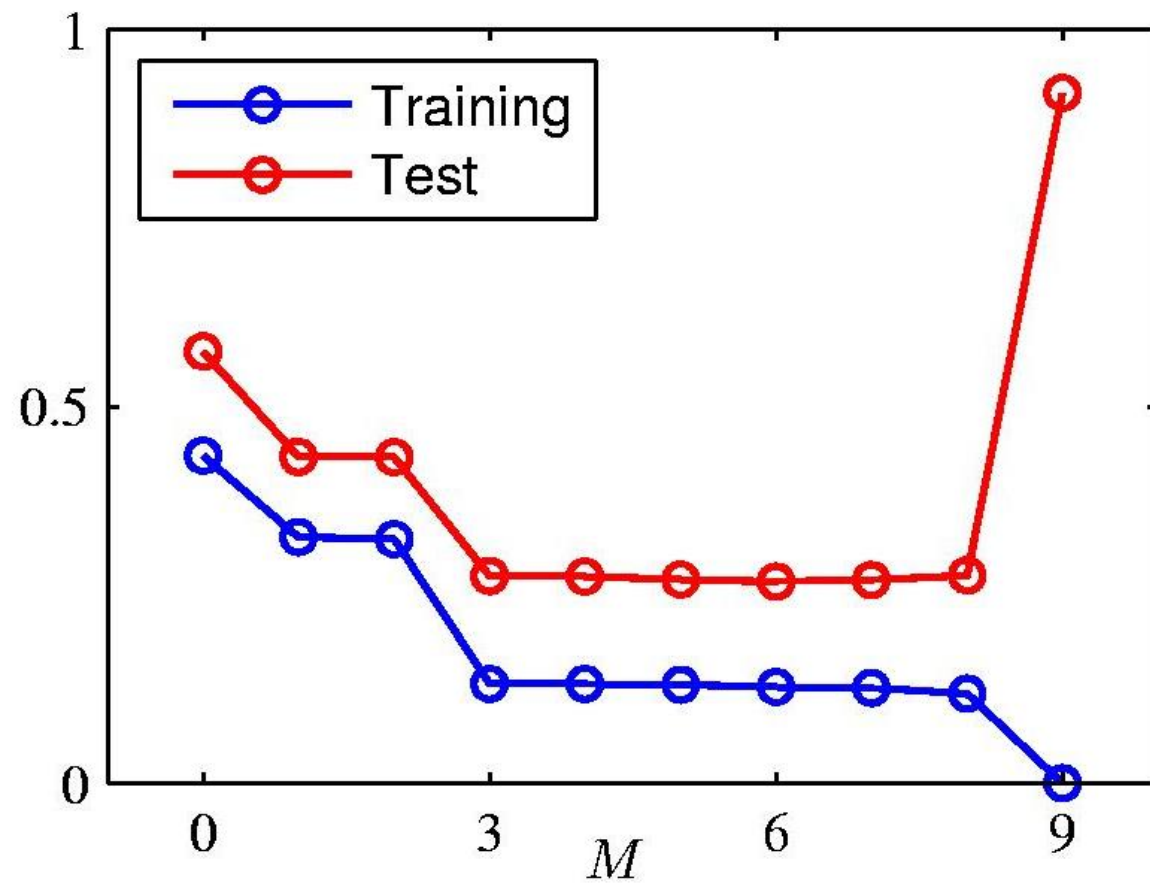
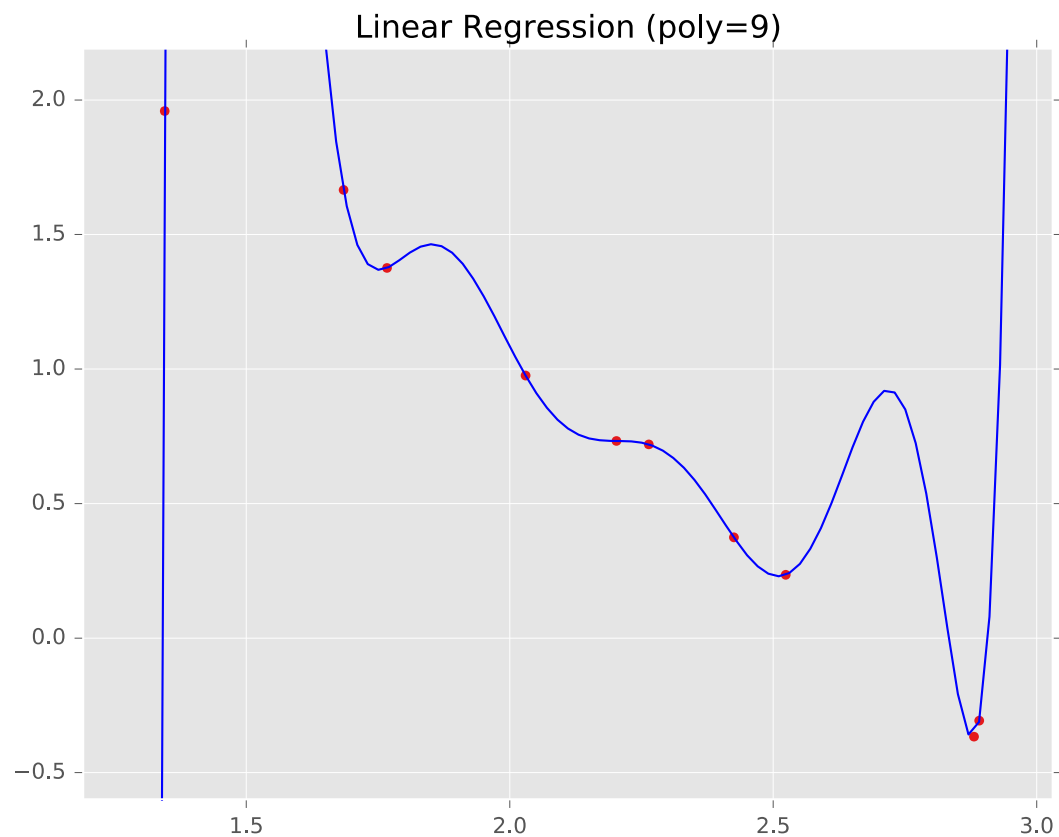
<https://www.youtube.com/watch?v=3liCbRZPrZA>

# Feature Maps

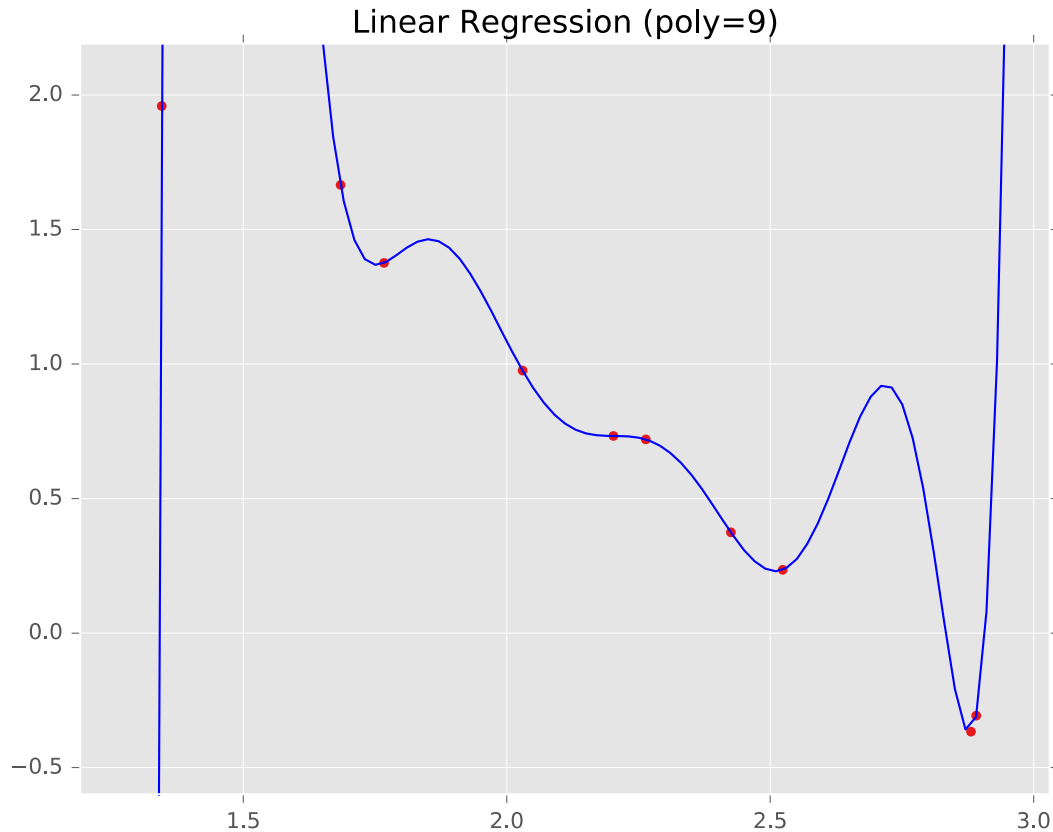
Two ways to think about it



# Overfitting: Symptoms



# Overfitting: Symptoms



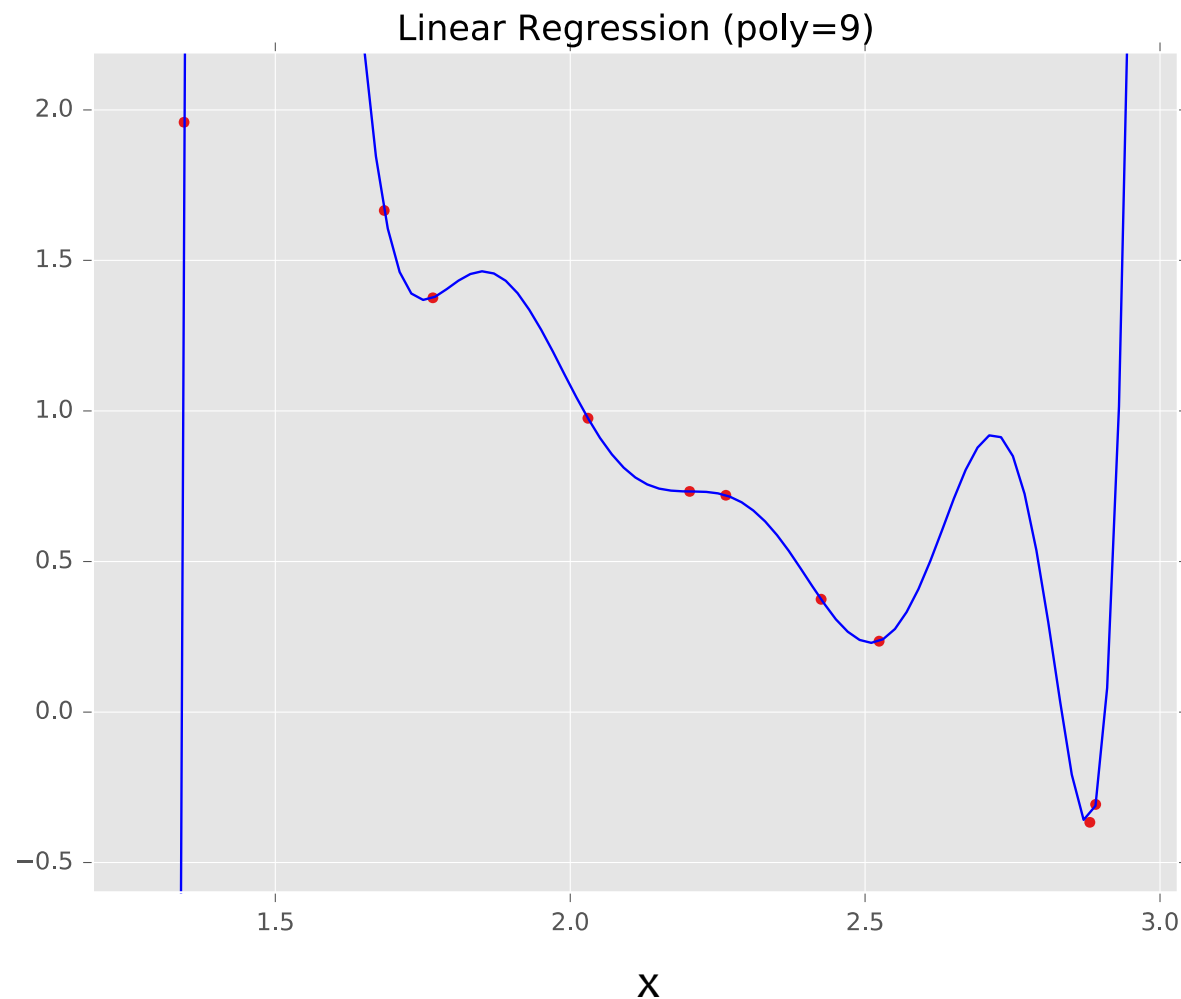
|            | $M = 0$ | $M = 1$ | $M = 3$ | $M = 9$     |
|------------|---------|---------|---------|-------------|
| $\theta_0$ | 0.19    | 0.82    | 0.31    | 0.35        |
| $\theta_1$ |         | -1.27   | 7.99    | 232.37      |
| $\theta_2$ |         |         | -25.43  | -5321.83    |
| $\theta_3$ |         |         | 17.37   | 48568.31    |
| $\theta_4$ |         |         |         | -231639.30  |
| $\theta_5$ |         |         |         | 640042.26   |
| $\theta_6$ |         |         |         | -1061800.52 |
| $\theta_7$ |         |         |         | 1042400.18  |
| $\theta_8$ |         |         |         | -557682.99  |
| $\theta_9$ |         |         |         | 125201.43   |

# Overfitting: Fix?

**Goal:** Learn  $y = \mathbf{w}^T \mathbf{f}(\mathbf{x}) + b$   
where  $\mathbf{f}(\cdot)$  is a polynomial  
basis function

| i   | y   | x   | ... | $x^9$     |
|-----|-----|-----|-----|-----------|
| 1   | 2.0 | 1.2 | ... | $(1.2)^9$ |
| 2   | 1.3 | 1.7 | ... | $(1.7)^9$ |
| ... | ... | ... | ... | ...       |
| 10  | 1.1 | 1.9 | ... | $(1.9)^9$ |

y

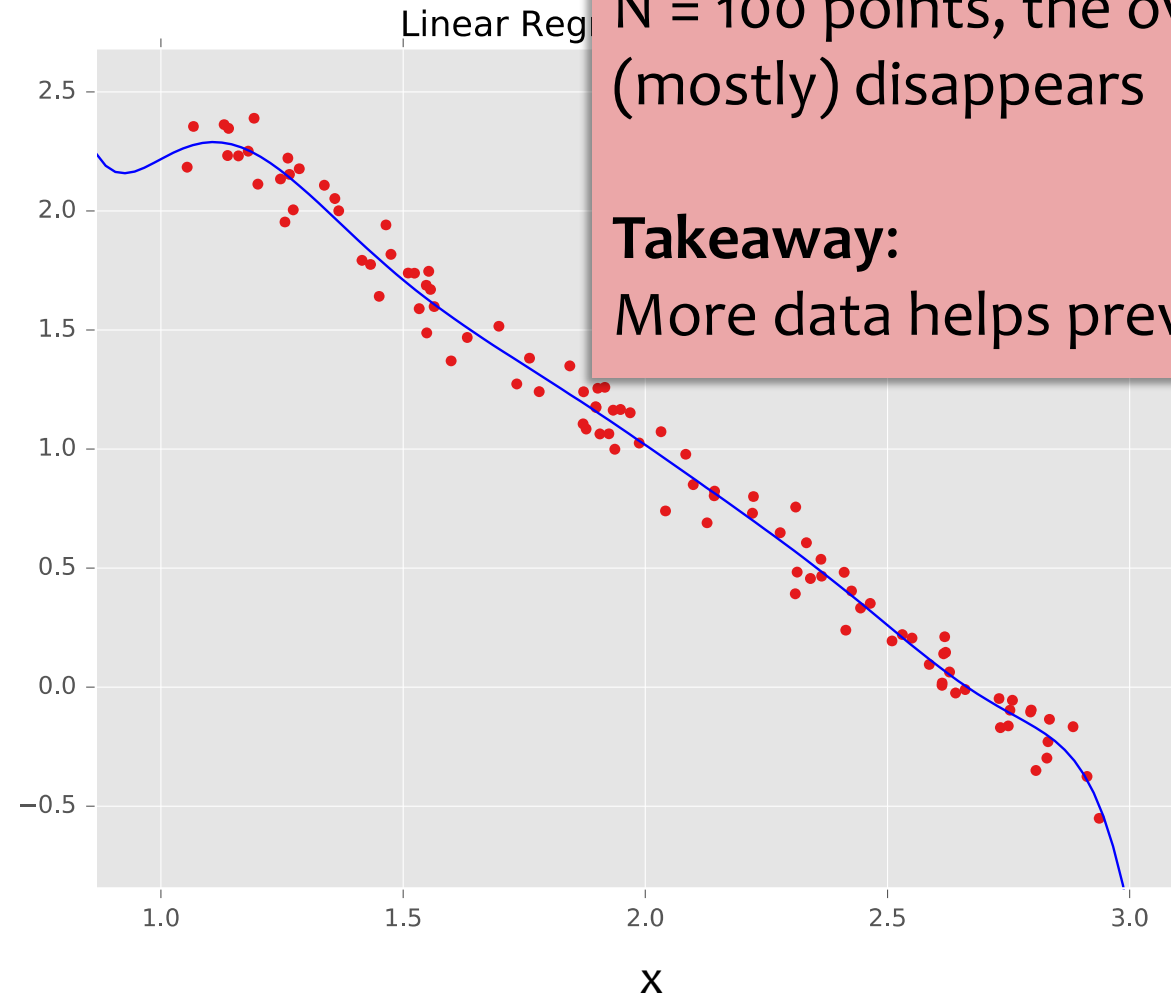


N = 10 points we overfit!

# Overfitting: More training data!

**Goal:** Learn  $y = \mathbf{w}^T \mathbf{f}(\mathbf{x}) + b$   
where  $\mathbf{f}(\cdot)$  is a polynomial  
basis function

| i   | y   | x   | ... | $x^9$     |
|-----|-----|-----|-----|-----------|
| 1   | 2.0 | 1.2 | ... | $(1.2)^9$ |
| 2   | 1.3 | 1.7 | ... | $(1.7)^9$ |
| 3   | 0.1 | 2.7 | ... | $(2.7)^9$ |
| 4   | 1.1 | 1.9 | ... | $(1.9)^9$ |
| ... | ... | ... | ... | ...       |
| ... | ... | ... | ... | ...       |
| ... | ... | ... | ... | ...       |
| 98  | ... | ... | ... | ...       |
| 99  | ... | ... | ... | ...       |
| 100 | 0.9 | 1.5 | ... | $(1.5)^9$ |



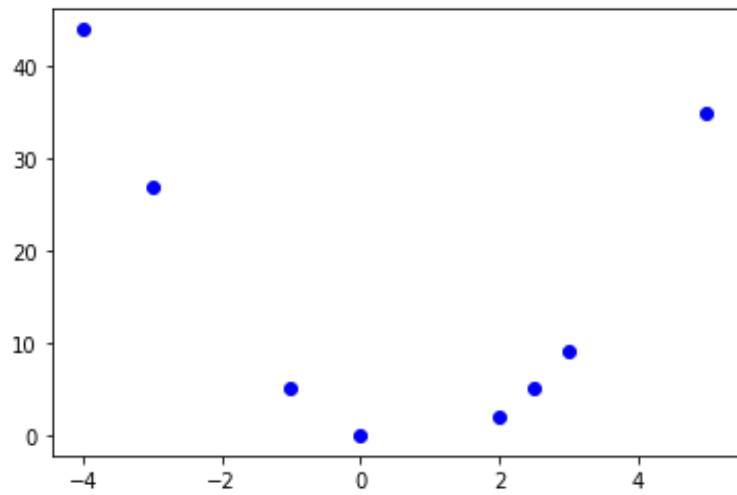
$N = 10$  points we overfit!

$N = 100$  points, the overfitting  
(mostly) disappears

**Takeaway:**  
More data helps prevent overfitting

# Polynomial Features

## Design Matrix



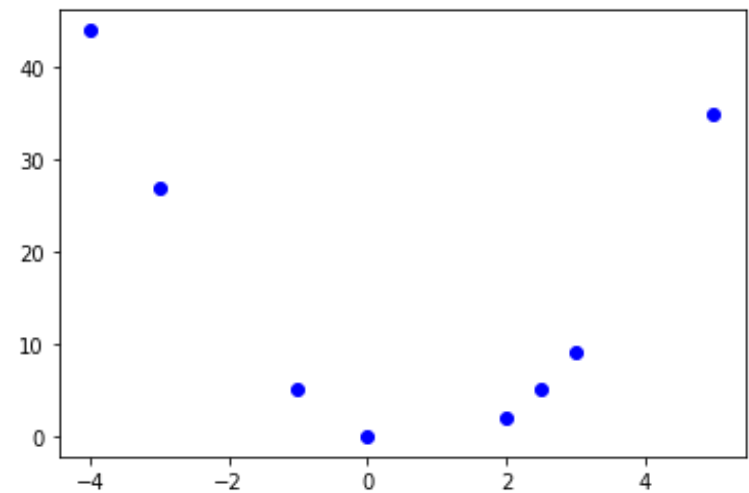
| $y$ | $x_1$ |
|-----|-------|
| 44  | -4    |
| 27  | -3    |
| 5   | -1    |
| 0   | 0     |
| 2   | 2     |
| 5   | 2.5   |
| 9   | 3     |
| 35  | 5     |

$$X = \begin{bmatrix} \text{---} \mathbf{x}^{(1)\top} \text{---} \\ \text{---} \mathbf{x}^{(2)\top} \text{---} \\ \text{---} \mathbf{x}^{(3)\top} \text{---} \\ \vdots \\ \text{---} \mathbf{x}^{(N)\top} \text{---} \end{bmatrix}$$

$$\phi(X) = \begin{bmatrix} \text{---} \phi(\mathbf{x}^{(1)})^\top \text{---} \\ \text{---} \phi(\mathbf{x}^{(2)})^\top \text{---} \\ \text{---} \phi(\mathbf{x}^{(3)})^\top \text{---} \\ \vdots \\ \text{---} \phi(\mathbf{x}^{(N)})^\top \text{---} \end{bmatrix}$$

# Polynomial Features

Solving linear regression



| $y$ | $x_1$ |
|-----|-------|
| 44  | -4    |
| 27  | -3    |
| 5   | -1    |
| 0   | 0     |
| 2   | 2     |
| 5   | 2.5   |
| 9   | 3     |
| 35  | 5     |

# Feature Engineering vs Feature Learning

## Feature engineering

- Humans decide what the features may be useful
- Humans implement feature extraction algorithms

## Feature learning

- Humans choose data and performance measure
- Humans decide on structure/algorithm to learn features
  - Features are just intermediate values between input and output
  - Structure defines number of feature values
- Allow machine to map data to feature values as needed

# One-hot encoding

Need to convert text for categories into numerical values for input and output

## Categorical variables

- Enumeration
- One-hot vector (or one-hot columns)

## One-hot vector

- Vector of length  $M$  with exactly one 1 and  $M-1$  zeros
- Series of  $M$  binary values, exactly one of which is 1

Categorical

| <u>color</u> | <u>color</u> | <u>r</u> | <u>g</u> | <u>b</u> |
|--------------|--------------|----------|----------|----------|
| red          | 1            | 1        | 0        | 0        |
| green        | 2            | 0        | 1        | 0        |
| blue         | 3            | 0        | 0        | 1        |
| red          | 1            | 1        | 0        | 0        |
| red          | 1            | 1        | 0        | 0        |

# More to come later in the semester

✓ Polynomial feature engineering

✓ Categorical features and one-hot encoding

## Images

- Raw data, feature engineering
- Feature learning

## Audio

- Raw data, feature engineering
- Feature learning

## Text

- Raw data, feature engineering
- Feature learning