Bias-Variance Tradeoff and Regularization

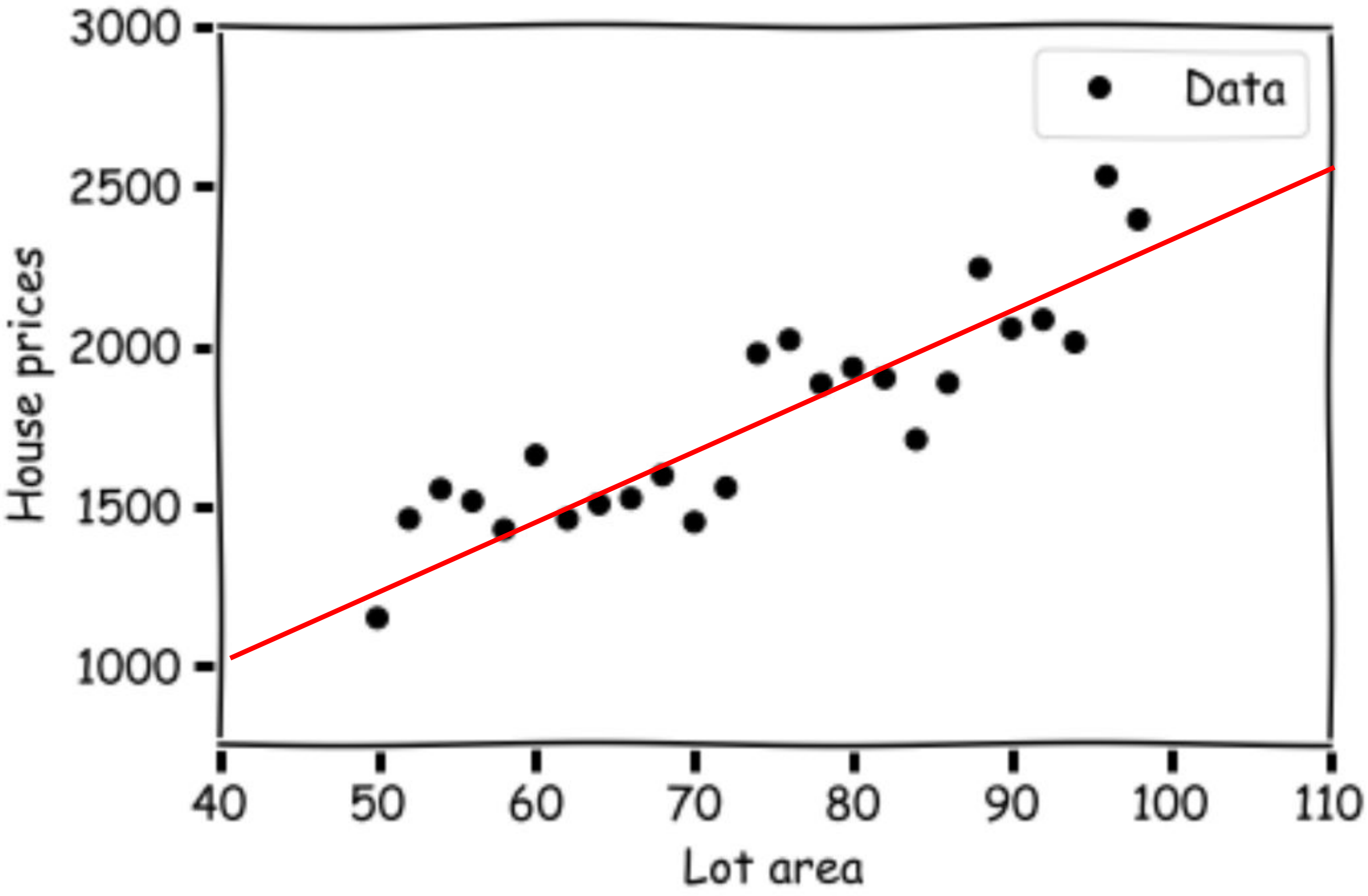
Original Slides by:
Courtney Anne Ngo
Daniel Stanley Tan, PhD
Arren Antioquia

Updated (AY 2023 – 2024 T3) by:
Thomas James Tiam-Lee, PhD

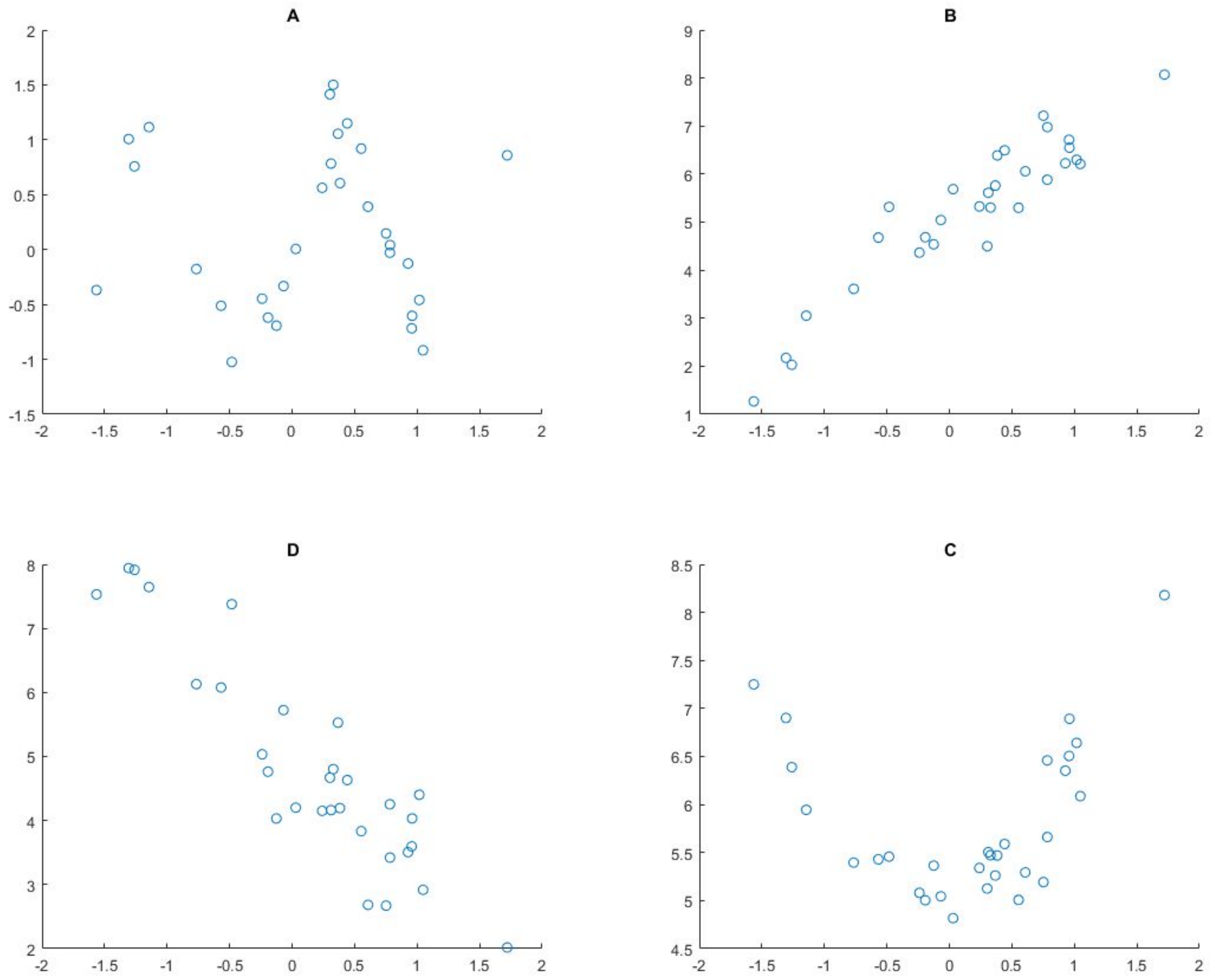
DE LA SALLE UNIVERSITY
RELIGIO MORES CULTURA
MANILA

College of
Computer Studies

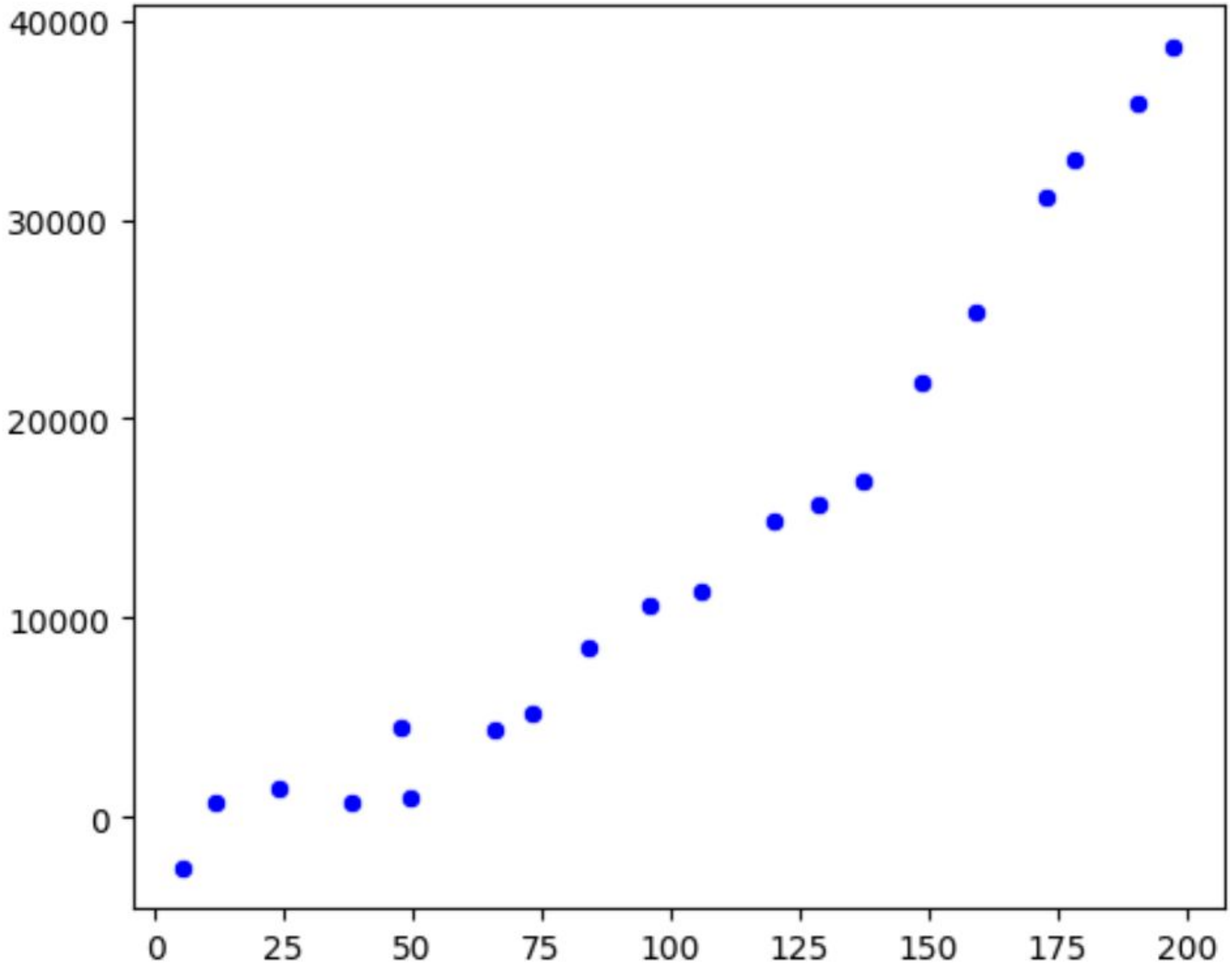
Linear Regression



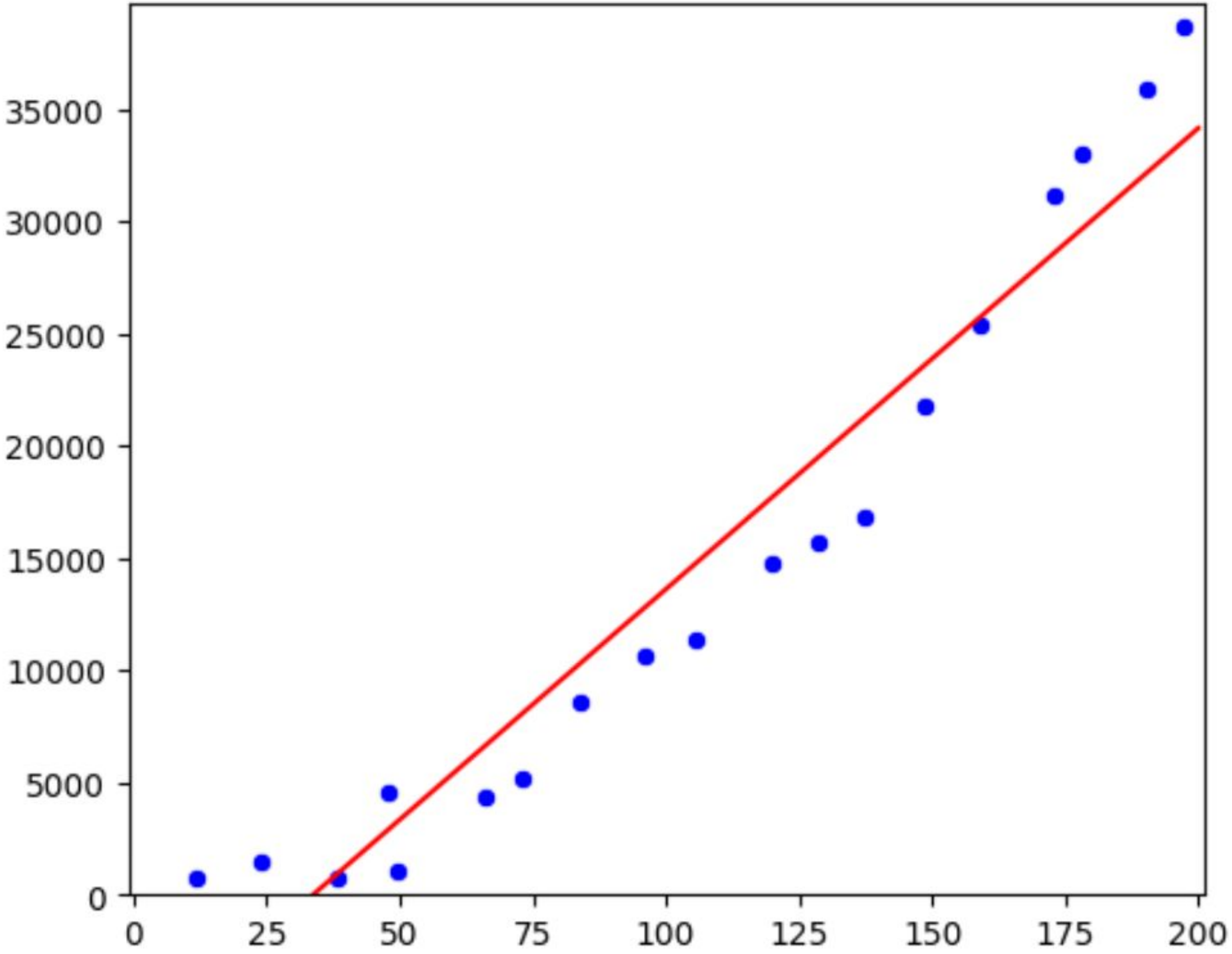
- Which of these data will linear regression be suitable in?



The Data



The Data



- Fitting a standard linear regression model

$$\hat{y} = w_1 x + w_0$$

- Can we do better?

Alternative Model

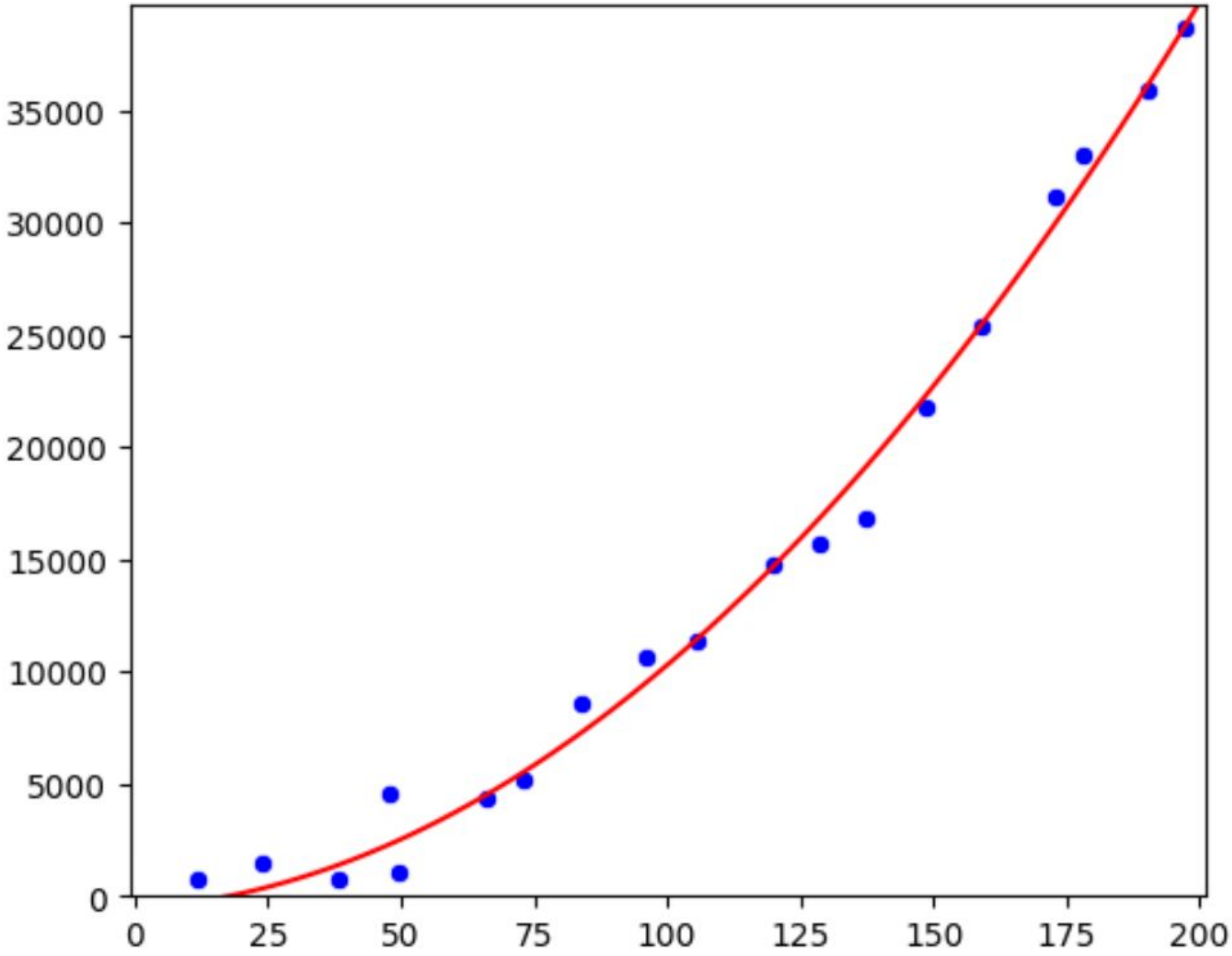
$$\hat{y} = \theta_1 x + \theta_0$$

$$\Downarrow$$

$$\hat{y} = \theta_1 x^2 + \theta_0$$

- What happens to the model?

The Data



- Fitting the updated model

$$\hat{y} = \theta_1 x^2 + \theta_0$$

- Do you think it's a better model?

So Is It Still "Linear" Regression (? 🤔)

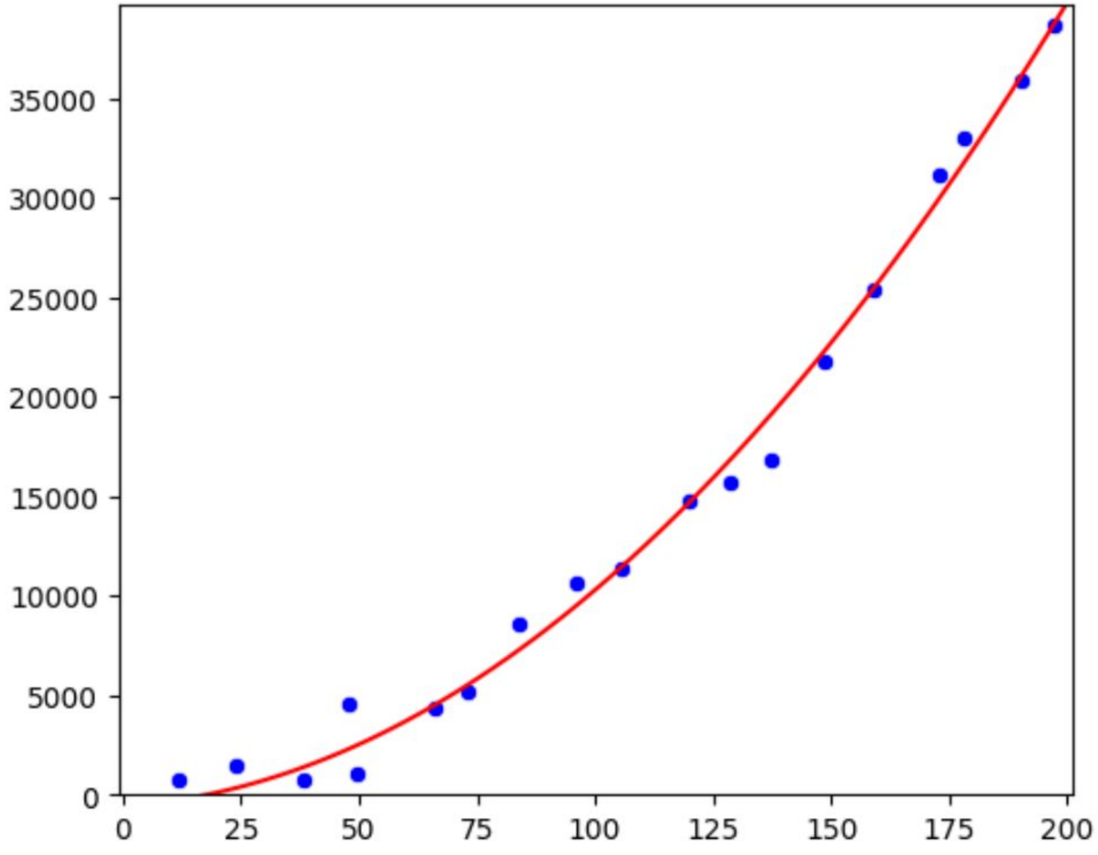
- There's two ways to frame what we did:
  - ① **We change the model by changing x to x^2**

Original feature (x)

$$\hat{y} = \theta_1 x^2 + \theta_0$$

Model

Prediction ($\hat{y}$)



So Is It Still "Linear" Regression (?🤔)

- There's two ways to frame what we did:
  - **② We keep the model, but change the features**

Original feature (x)

↓

Transformed feature (x^2)

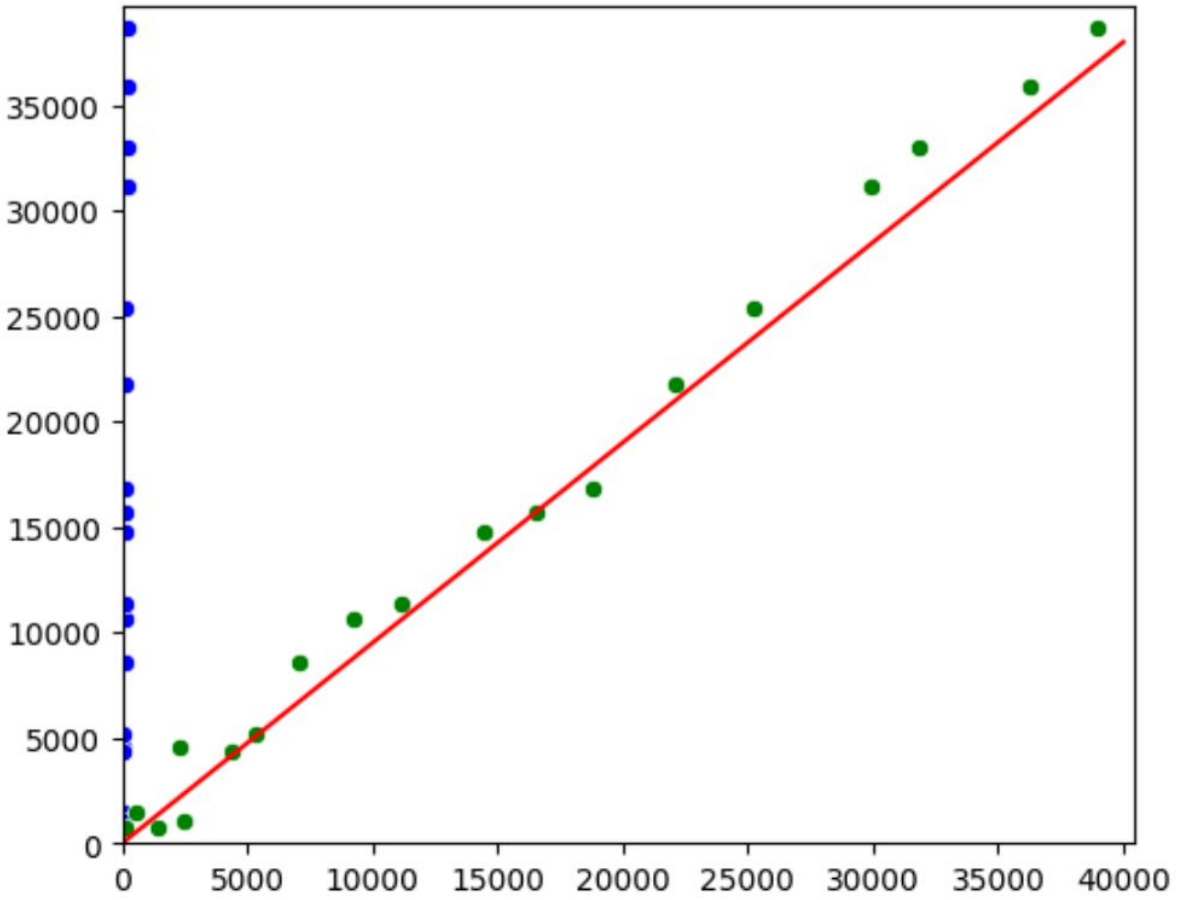
↓

$$\boxed{\hat{y} = \theta_1 x + \theta_0}$$

Model

↓

Prediction ($\hat{y}$)



Taking It a Step Further (1 feature)…

- **Order = 1**
  - Features: x_1
  - Model: $\hat{y} = \theta_1 x_1 + \theta_0$
- **Order = 2**
  - Features: x_1, x_1^2
  - Model: $\hat{y} = \theta_1 x_1 + \theta_2 x_1^2 + \theta_0$
- **Order = 3**
  - Features: x_1, x_1^2, x_1^3
  - Model: $\hat{y} = \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^3 + \theta_0$

Taking It a Step Further (2 features)…

- **Order = 1**
  - Features: x_1, x_2
  - Model: $\hat{y} = \theta_1 x_1 + \theta_2 x_2 + \theta_0$
- **Order = 2**
  - Features: $x_1, x_2, x_1^2, x_1 x_2, , x_2^2,$
  - Model: $\hat{y} = \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_1 x_2 + \theta_5 x_2^2 + \theta_0$
- **Order = 3**
  - Features:$x_1, x_2, x_1^2, x_1 x_2, x_2^2, x_1^3, x_1^2 x_2, x_1 x_2^2, x_2^3$
  - Model: $\hat{y} = \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_1 x_2 + \theta_5 x_2^2 + \theta_6 x_1^3 + \theta_7 x_1^2 x_2 + \theta_8 x_1 x_2^2 + \theta_9 x_2^3 + \theta_0$

Curse of DIMENSIONALITY

As the dimensionality of the features space increases, the number configurations can grow exponentially, and thus the number of configurations covered by an observation decreases.

Chris Albon

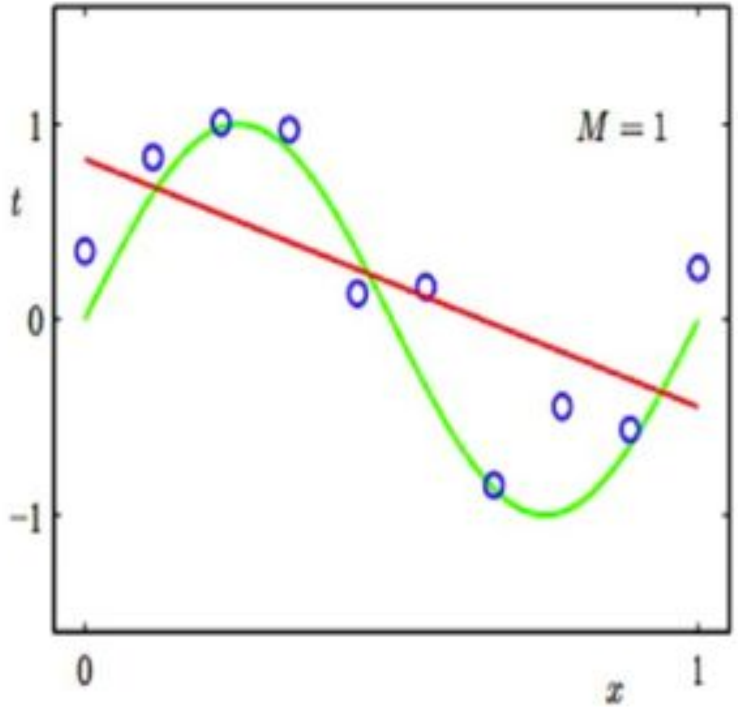
Image source: https://builtin.com/data-science/curse-dimensionality

Another Way to Do It (2 features)

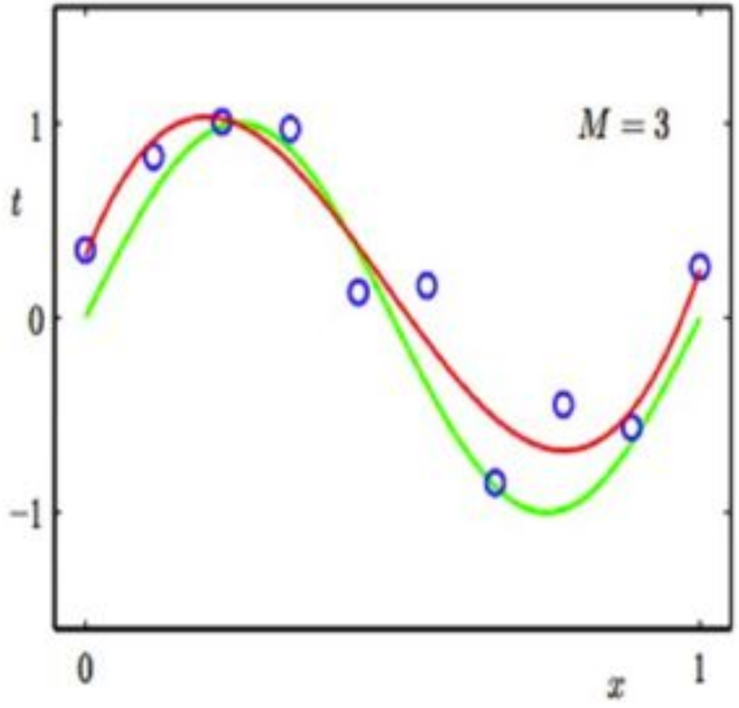
- **Order = 1**
  - Features: x_1, x_2
  - Model: $\hat{y} = \theta_1 x_1 + \theta_2 x_2 + \theta_0$
- **Order = 2**
  - Features: $x_1, x_2, x_1^2, x_2^2,$
  - Model: $\hat{y} = \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_0$
- **Order = 3**
  - Features:$x_1, x_2, x_1^2, x_2^2, x_1^3, x_2^3$
  - Model: $\hat{y} = \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1^3 + \theta_6 x_2^3 + \theta_0$

What Happens with Higher Order Models?

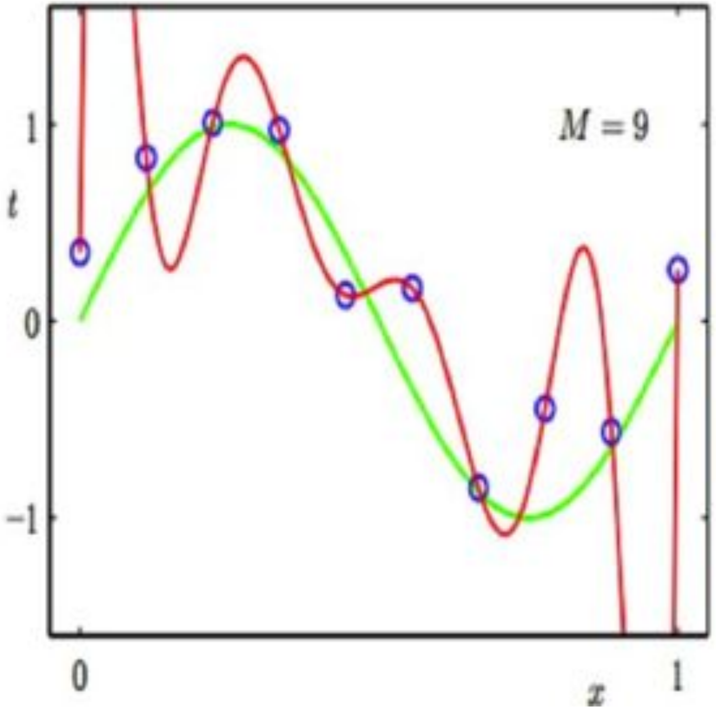
Order = 1



Order = 3

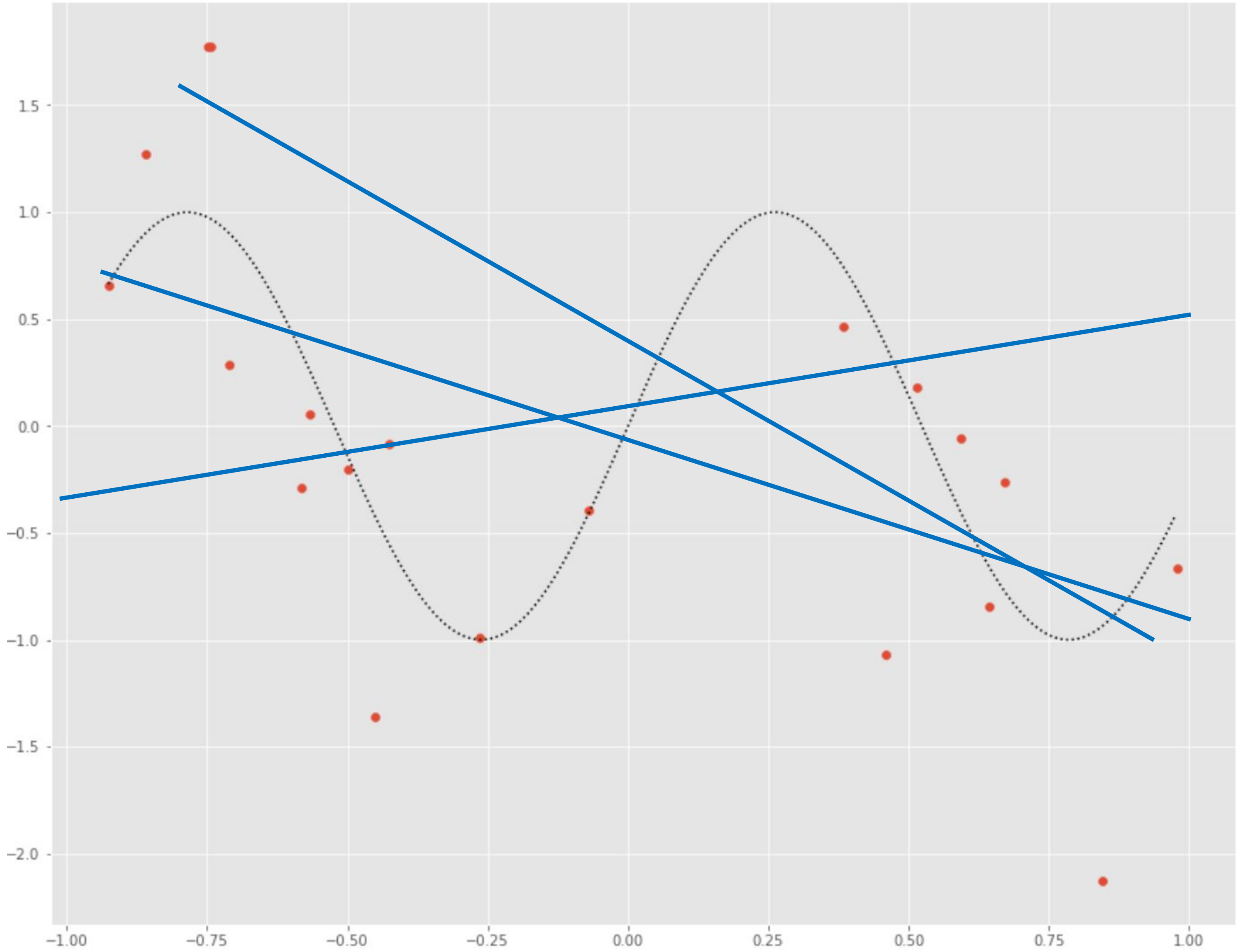


Order = 9



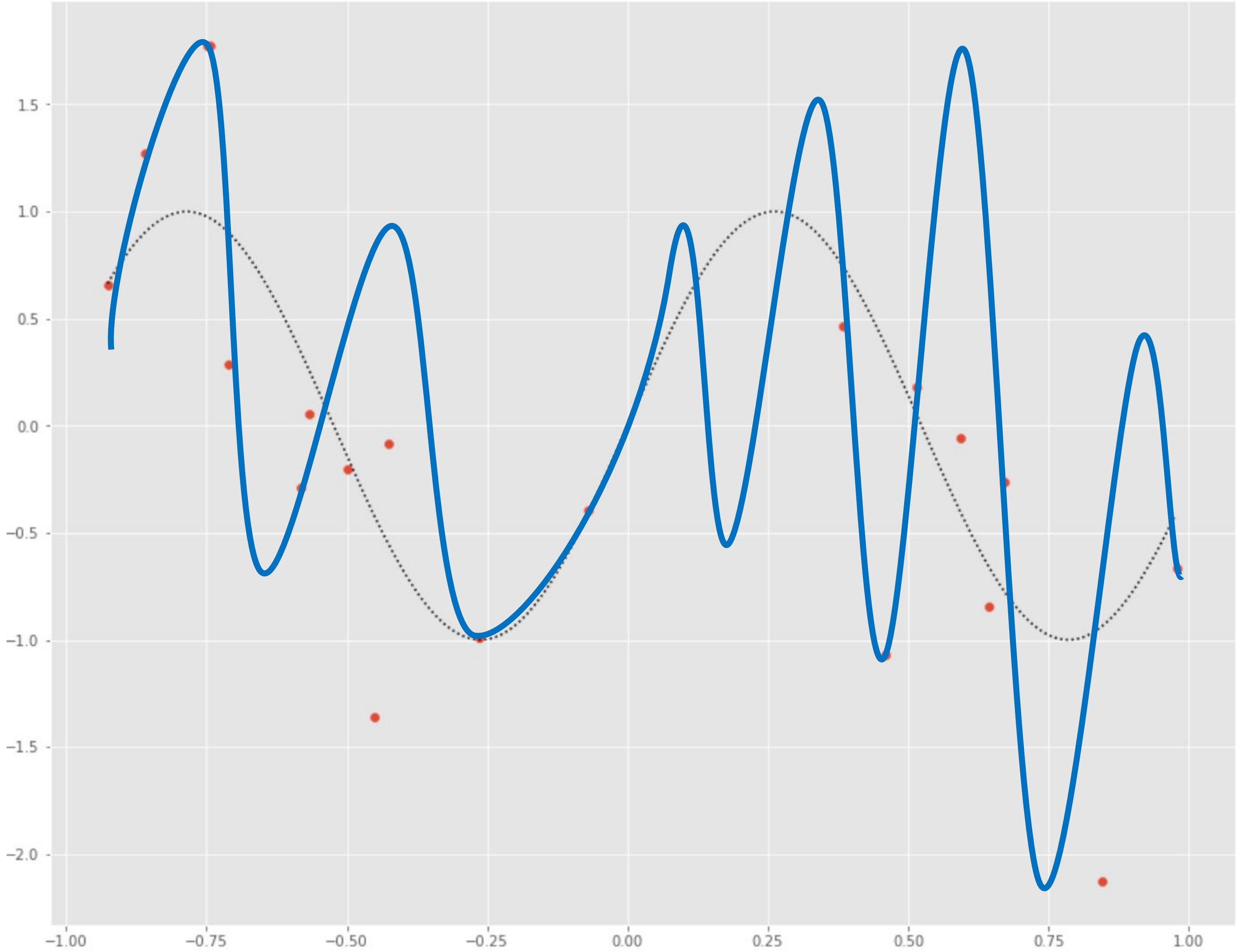
High Bias

- Model is too simple
- No matter how much you try to fit, it won’t capture the patterns of the data



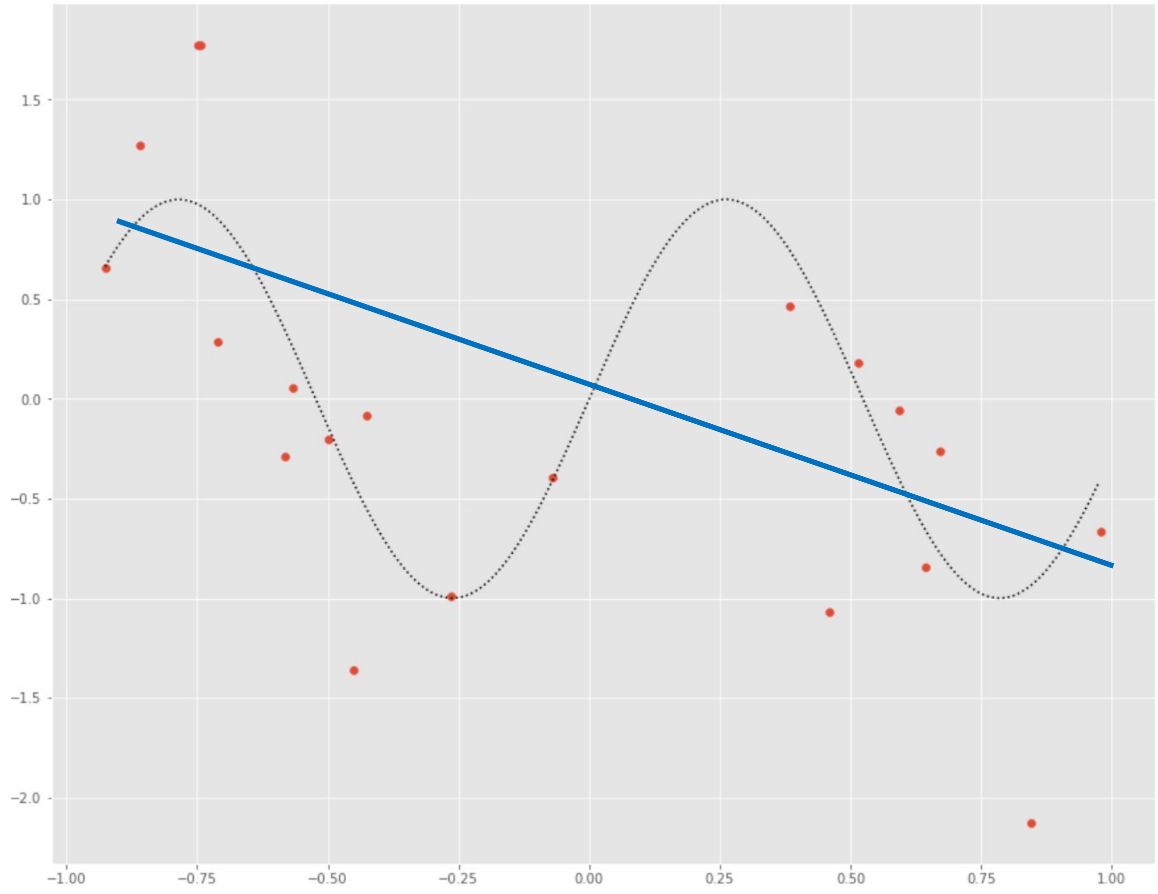
High Variance

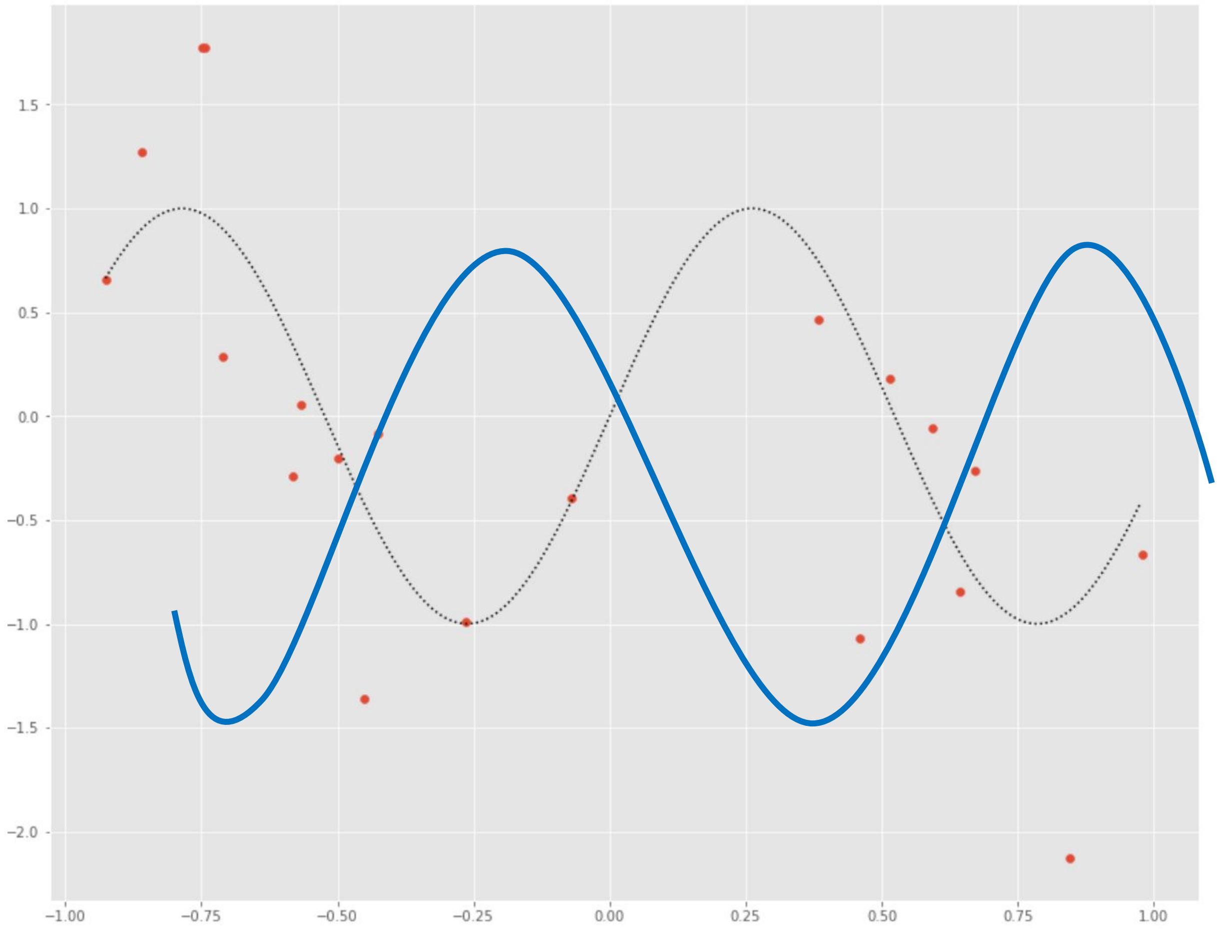
- Model is too complex
- A large variety of models with the same complexity can fit the data just as nicely



Underfitting

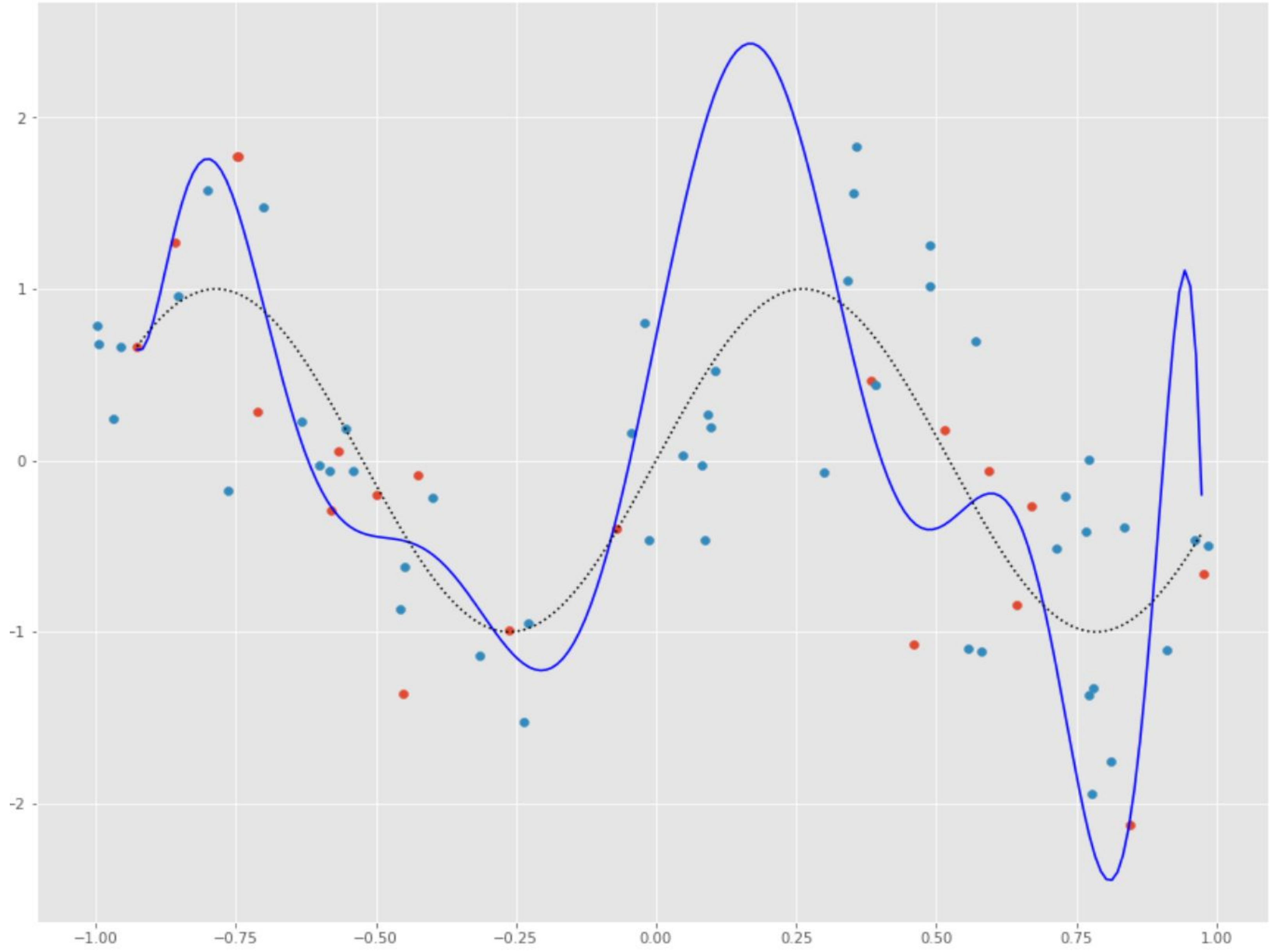
- Model did not fit the **training** data well.

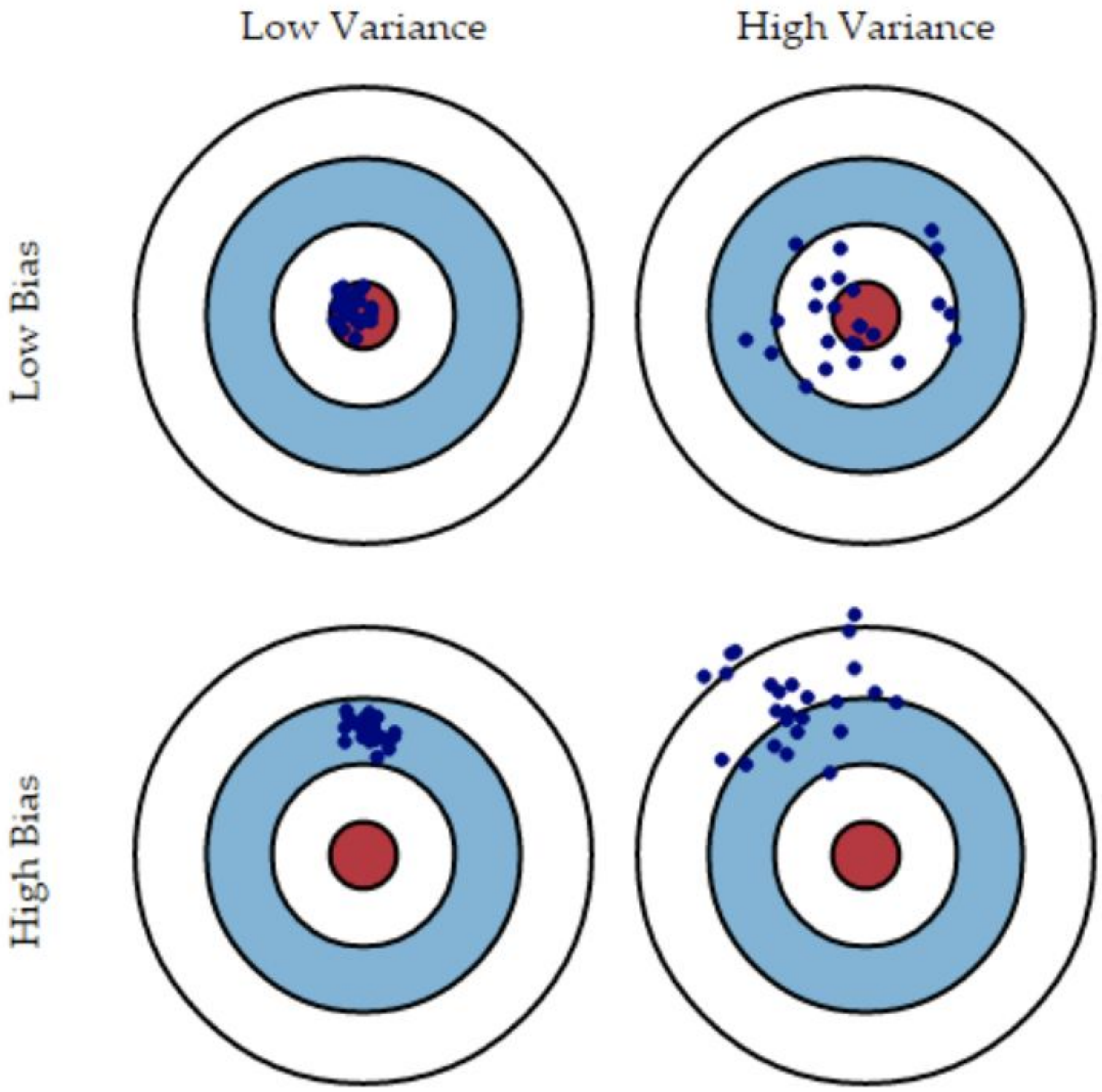




Overfitting

- Model fits the **training** data well, but performs poorly on the **testing** data, i.e. did not generalize

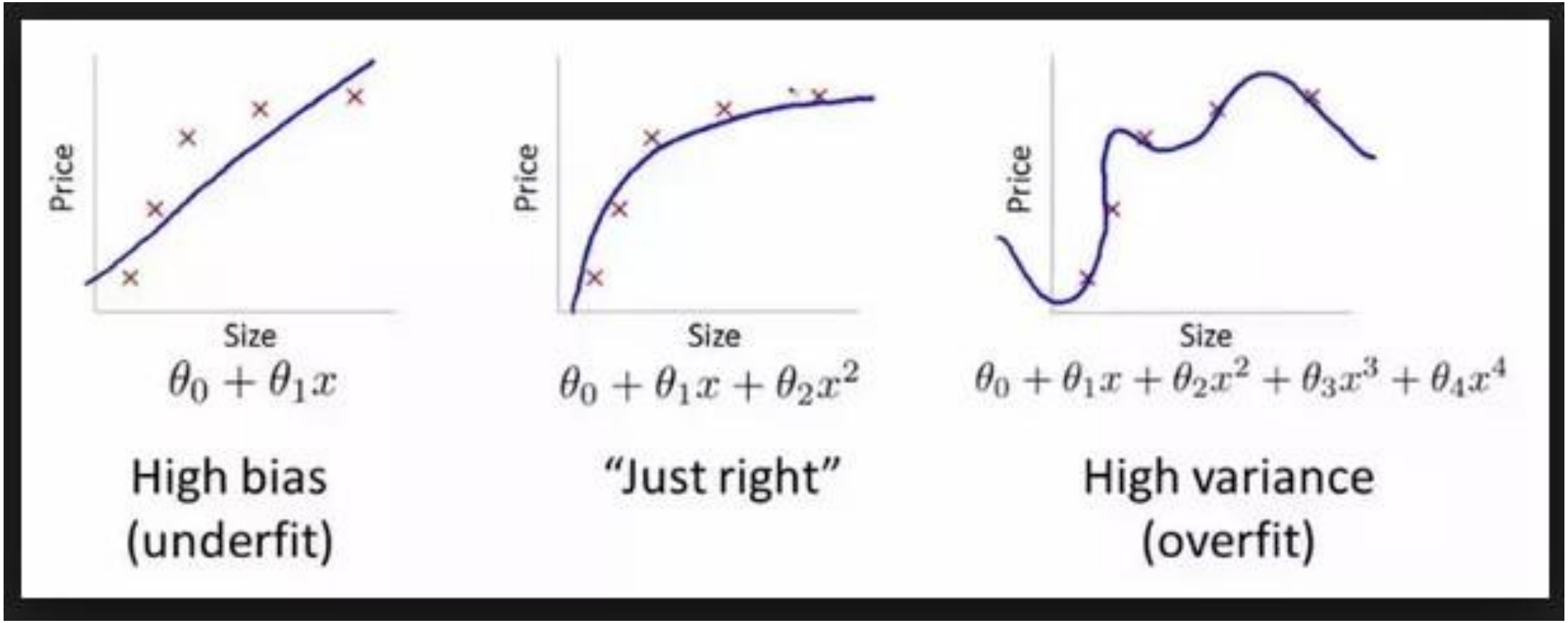



Low Variance
High Variance
Low Bias
High Bias

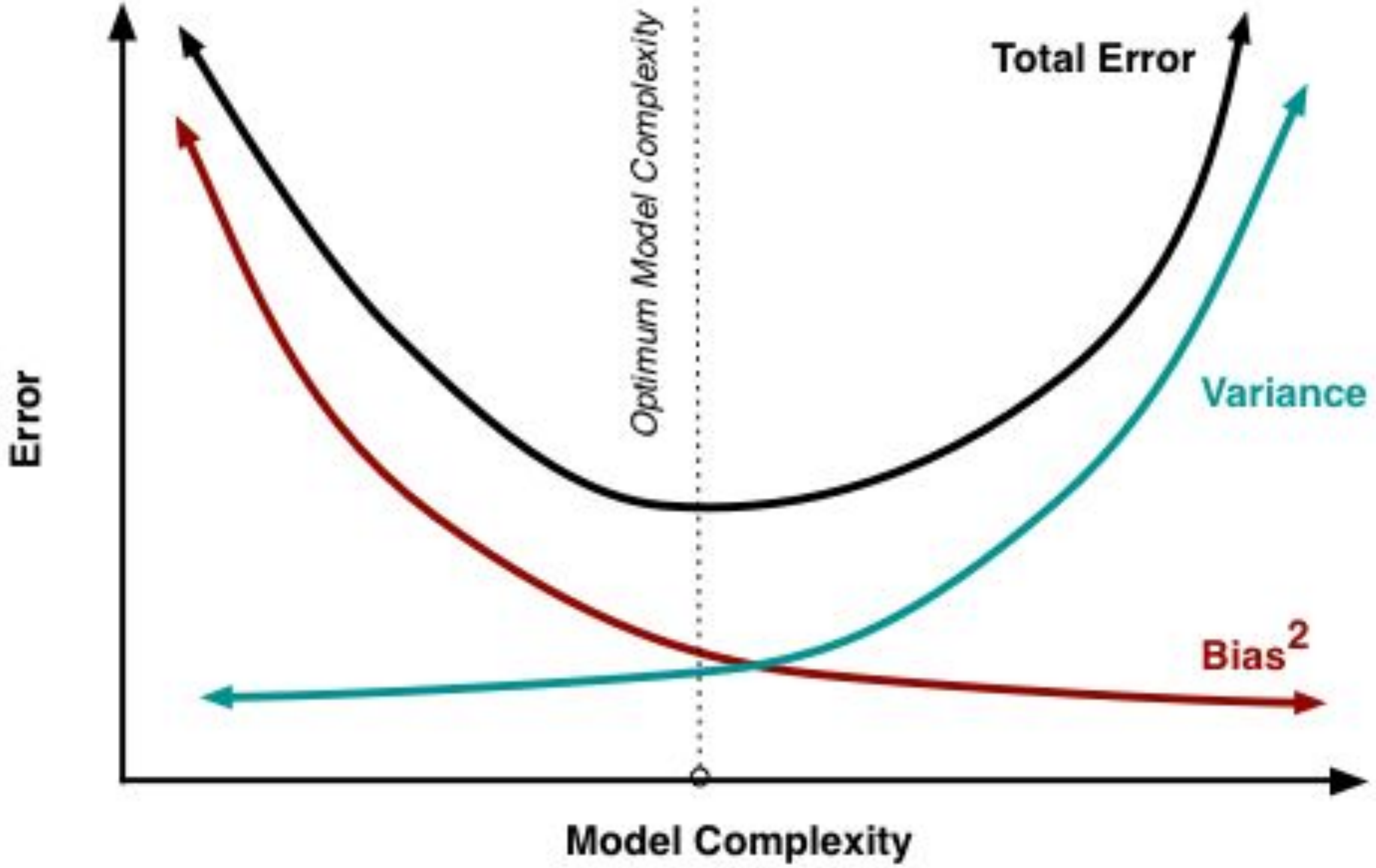
	Training error	*Test error*
Underfit model	**HIGH**	**HIGH**
Overfit model	**LOW**	**HIGH**

Bias-Variance Tradeoff

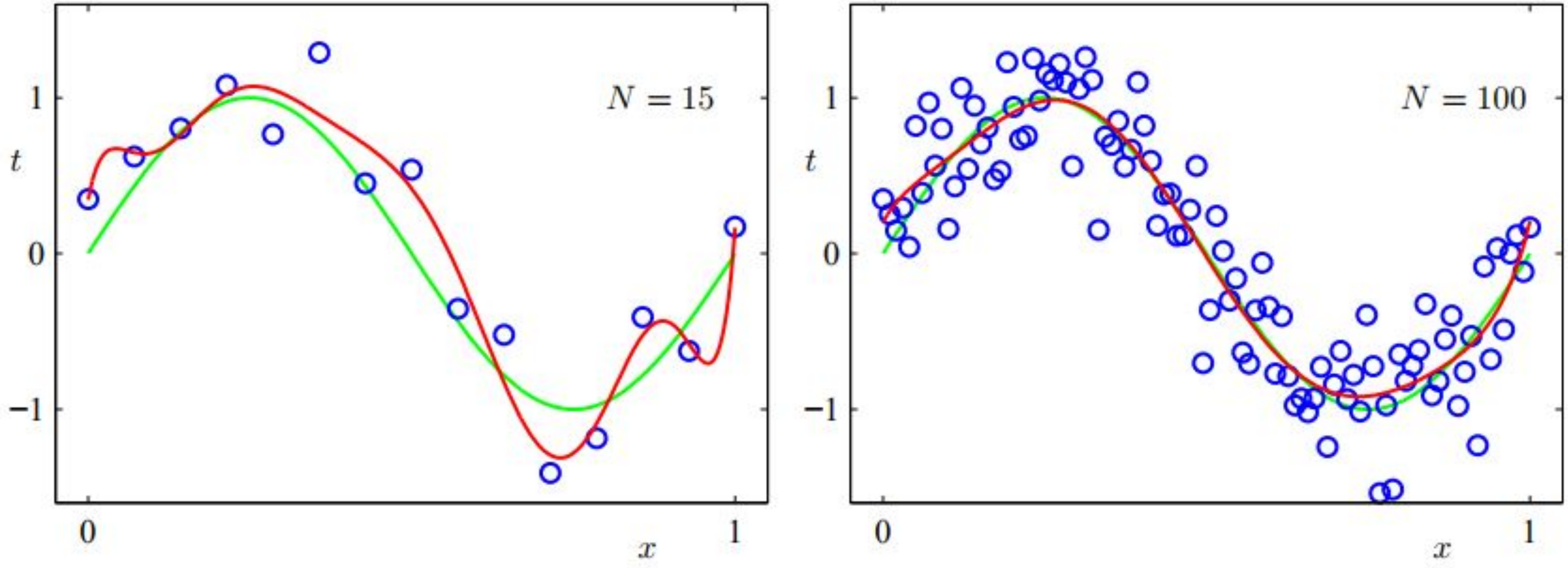
- Generally, we want to find a good balance between the bias and the variance.



Bias-Variance Tradeoff



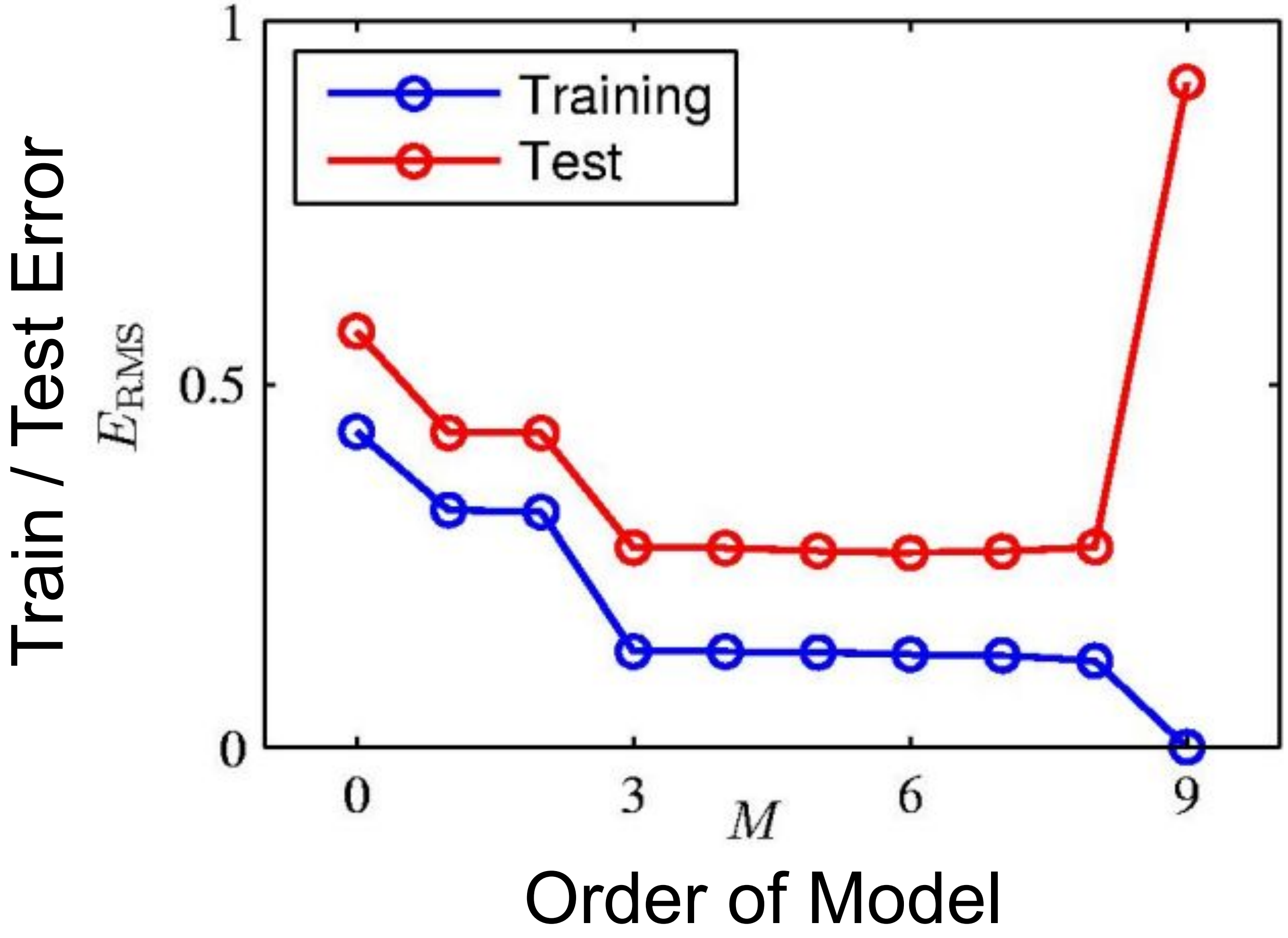
Effect of Data Quantity to Bias/Variance



More data means the models are more likely to “listen” to the general trend

Typical Overfitting Plot

- The training error decreases as the degree of the polynomial increases (i.e. complexity of the hypothesis)
- The testing error, measured on independent data, decreases at first, then starts increasing.



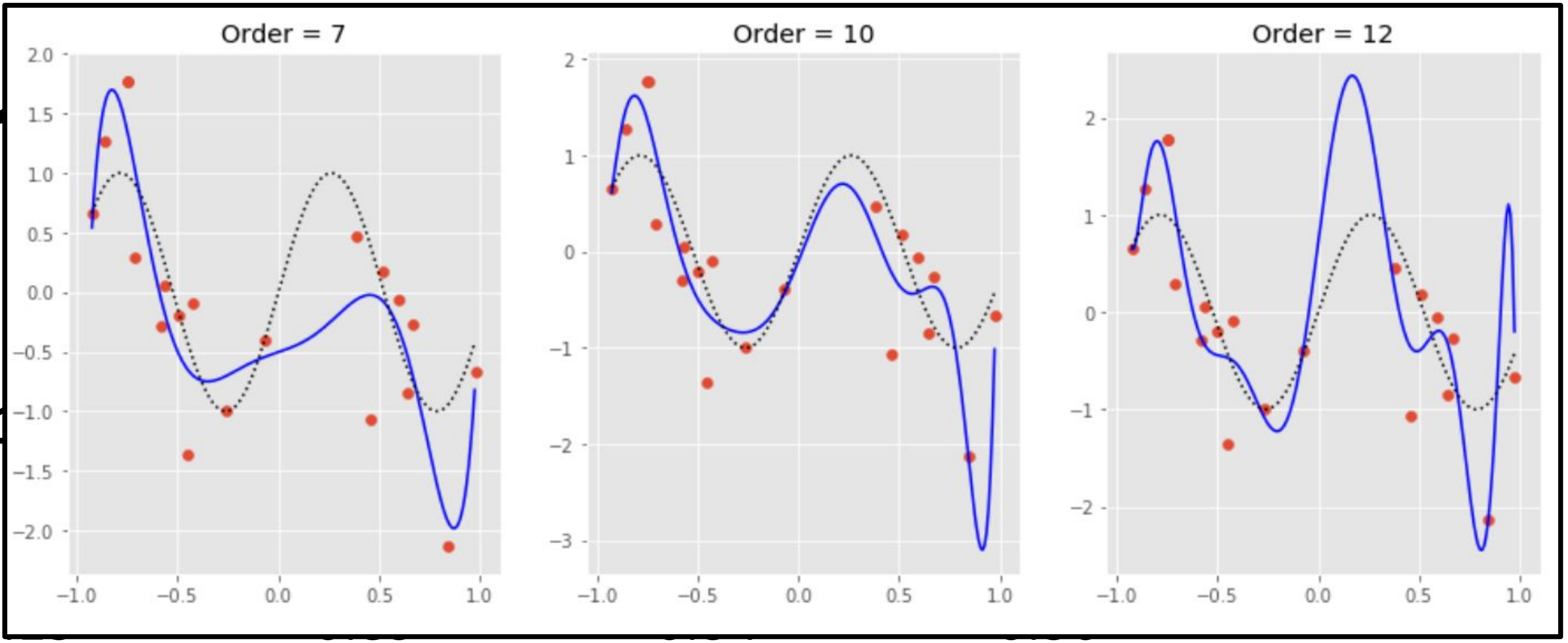
Regularization

- Methods to **reduce overfitting** in machine learning models, without having to collect more data or change the learning algorithm.

What Happens When We Increase the Order of the Polynomial?

poly degree	weight_1	weight_2	weight_3	weight_4	weight_5
1	-0.87	-0.17			
3	-2.15	0.89	0.34	-0.50	
5	13.40	-1.31	-17.28	1.36	3.79
7	31.30	-5.53	-37.29	4.48	6.87
10	179.74	94.89	-378.87	-185.43	259.32
12	-540.45	-1015.67	1487.13	2623.60	-1490.51
30	-14156072.98	16329720.76	21300299.24	-14967132.99	13066492.56

What Happen
Order of the



Wild swings are being caused by large magnitude weights!

poly degree	weight_1				
1	-0				
3	-2				
5					
7	31.30	-5.53	-37.29	4.48	6.87
10	179.74	94.89	-378.87	-185.43	259.32
12	-540.45	-1015.67	1487.13	2623.60	-1490.51
30	-14156072.98	16329720.76	21300299.24	-14967132.99	13066492.56

Linear Regression Loss Function

$$l(\theta, X, y) = \frac{1}{2n} \sum (\hat{y} - y)^2$$

Linear Regression Loss Function

- We add a **regularization term**:

$$l(\theta, X, y) = \underbrace{\frac{1}{2n}\sum(\hat{y} - y)^2}_{\text{Training error}} + \underbrace{\lambda R(\theta)}_{\text{Regularization error}}$$

Regularization Term

- Ridge Regression (L2 regularization):

$$R(\theta) = \sum_{j=1}^{d} \theta_j^2$$

- Lasso Regression (L1 regularization):

$$R(\theta) = \sum_{j=1}^{d} |\theta_j|$$

Regularized Linear Regression

- We add a **regularization term**:

$$l(\theta, X, y) = \underbrace{\frac{1}{2n}\sum(\hat{y} - y)^2}_{\text{Training error}} + \underbrace{\lambda \sum_{j=1}^{d} \theta_j^2}_{\text{Regularization error}}$$

Regularized Linear Regression

- We add a **regularization term**:

$$l(\theta, X, y) = \underbrace{\frac{1}{2n}\sum(\hat{y} - y)^2}_{\text{Training error } (\theta \text{ farther from } 0\text{, the better})} + \underbrace{\lambda \sum_{j=1}^{d} \theta_j^2}_{\text{Regularization error } (\theta \text{ nearer } 0\text{, the better})}$$

Regularized Linear Regression

- We add a **regularization term**:

Regularization constant

$$l(\theta, X, y) = \underbrace{\frac{1}{2n}\sum(\hat{y} - y)^2}_{\text{Training error } (\theta \text{ farther from } 0\text{, the better})} + \underbrace{\boxed{\lambda}\sum_{j=1}^{d} \theta_j^2}_{\text{Regularization error } (\theta \text{ nearer } 0\text{, the better})}$$

Effect of λ

$$l(\theta, X, y) = \frac{1}{2n} \sum (\hat{y} - y)^2 + \lambda \sum_{j=1}^{d} \theta_j^2$$

▪ If λ is large…	training error has little impact on loss
▪ If λ is 0…	regularization has no effect
▪ If λ is negative…	The higher the weights, the better (⁉️)
▪ If λ is small…	regularization is considered in the loss

Two Common Regularization Methods

- Ridge regression
  - (L2 regularization)

$$l(\theta, X, y) = \frac{1}{2n}\sum(\hat{y} - y)^2 + \frac{1}{2}\lambda\sum_{j=1}^{d}\theta_j^2$$

Gradient Descent Update Rule:

$$\theta := \theta - \alpha(\frac{1}{n}(X\theta - y)^T X + \lambda\theta)$$

- Lasso regression
  - (L1 regularization)

$$l(\theta, X, y) = \frac{1}{2n}\sum(\hat{y} - y)^2 + \lambda\sum_{j=1}^{d}|\theta_j|$$

Although absolute value is not differentiable, there are ways to generalize gradient descent to non-differentiable functions (will not be covered in this class)

poly degree	weight_1	weight_2	weight_3	weight_4	weight_5
1	-0.87	-0.17			
3	-2.15	0.89	0.34	-0.50	
5	13.40	-1.31	-17.28	1.36	3.79
7	31.30	-5.53	-37.29	4.48	6.87
10	179.74	94.89	-378.87	-185.43	259.32
12	-540.45	-1015.67	1487.13	2623.60	-1490.51
30	-14156072.98	16329720.76	21300299.24	-14967132.99	13066492.56

Effect of Ridge Regression

poly degree	weight_1	weight_2	weight_3	weight_4	weight_5	weight_6	weight_7	weight_8	weight_9	weight_10
1	-0.44									
3	0.17	0.22	-0.27							
5	1.90	0.34	-2.16	-0.08	0.42					
7	0.30	0.03	0.84	0.28	-1.16	-0.09	0.24			
10	1.78	-0.46	-3.06	-0.59	2.11	1.35	-0.82	-0.61	0.12	0.08
12	1.95	0.06	-3.43	-1.35	2.22	1.31	-0.70	-0.15	0.05	-0.12
20	2.78	-1.04	-6.02	1.21	4.23	-0.73	-0.68	0.15	-0.27	0.06

Effect of Lasso Regression

poly degree	weight_1	weight_2	weight_3	weight_4	weight_5	weight_6	weight_7	weight_8	weight_9	weight_10
1	-0.44									
3	0.17	0.22	-0.27							
5	1.90	0.34	-2.16	-0.08	0.42	0.00				
7	0.29	0.03	0.85	0.28	-1.16	-0.09	0.24			
10	2.44	0.57	-4.84	-3.22	3.63	3.49	-1.32	-1.30	0.17	0.16
12	4.99	-1.21	-15.19	0.00	18.06	0.60	-10.16	0.00	2.61	-0.13
20	3.97	-1.58	-9.05	-0.03	5.82	2.95	0.00	-2.38	-0.77	0.26

Ridge Vs. Lasso Regression