Lecture 6

Intro to machine learning

Julian Reif
Fall 2025

FIN 550 outline

- Module 1: Foundations - complete
  - Program in R
  - Deploy cloud computing services
  - Manage and visualize data

- Module 2: Machine learning - current
  - Use data and models to *predict* an outcome
  - Focus is on measuring correlations

- Module 3: Causal analysis - after midterm
  - Use data and models to estimate *causal* effects (econometrics)
  - Focus is on distinguishing between correlation and causation

What is a model?

> "Nothing is less real than realism. Details are confusing. It is only by selection, by elimination, by emphasis, that we get at the real meaning of things."
>
> --- Georgia O'Keefe

Red Canna (1915)

The world is complicated

- It is generally impossible to perfectly predict an outcome

- But, maybe, we can capture some features that improve our predictions

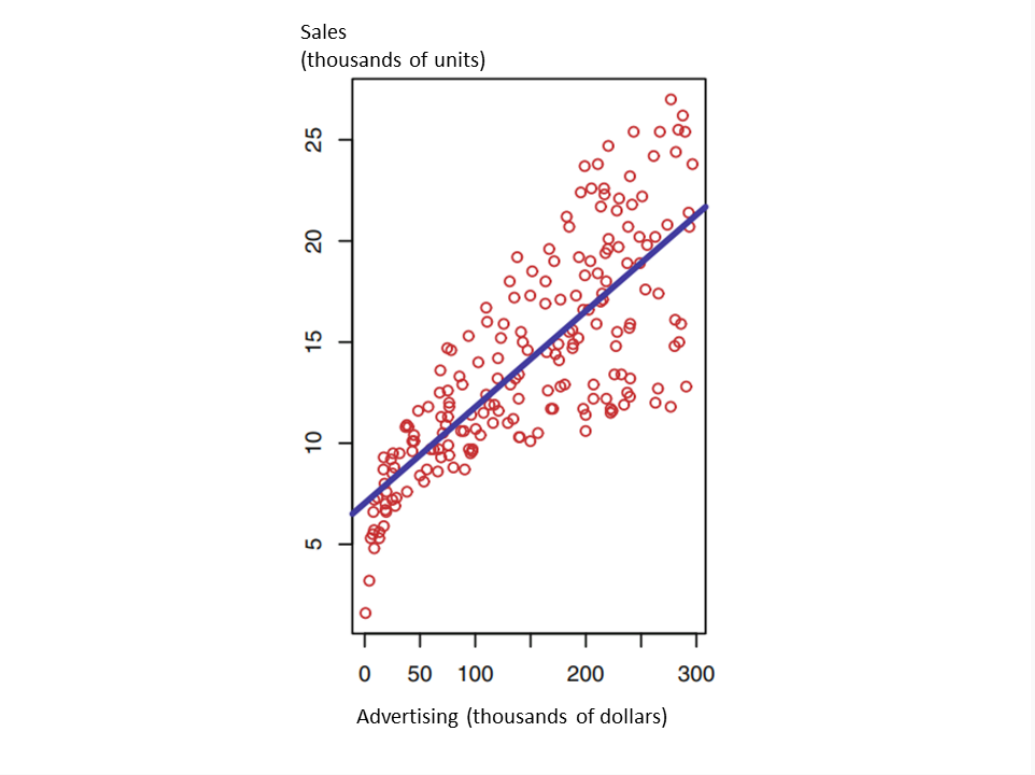
- Our ability to do that depends on:
  - Data availability
  - Statistical algorithms
  - Contextual knowledge

What determines a restaurant's sales?

- Product price
- Retail locations
- Advertising
- Weather
- Day of week
- Time of day
- Competitor's product price
- Competitor's retail locations
- How well a customer slept
- ... goes on and on!

Relationship between sales and advertising

- Make a scatter plot
  - Input data: advertising
  - Output data: sales
- Plot a line of best fit (linear regression)
- An increase in advertising systematically predicts an increase in sales



Machine learning (ML)

- ML is a computational approach to using data to form predictions

- Often classified into two types:
  1. Supervised learning: associate a set of measurements with a set of responses
     - X: input variables ("predictors/features/regressors" or "independent variables")
     - Y: output variable ("outcome/response/dependent variable")

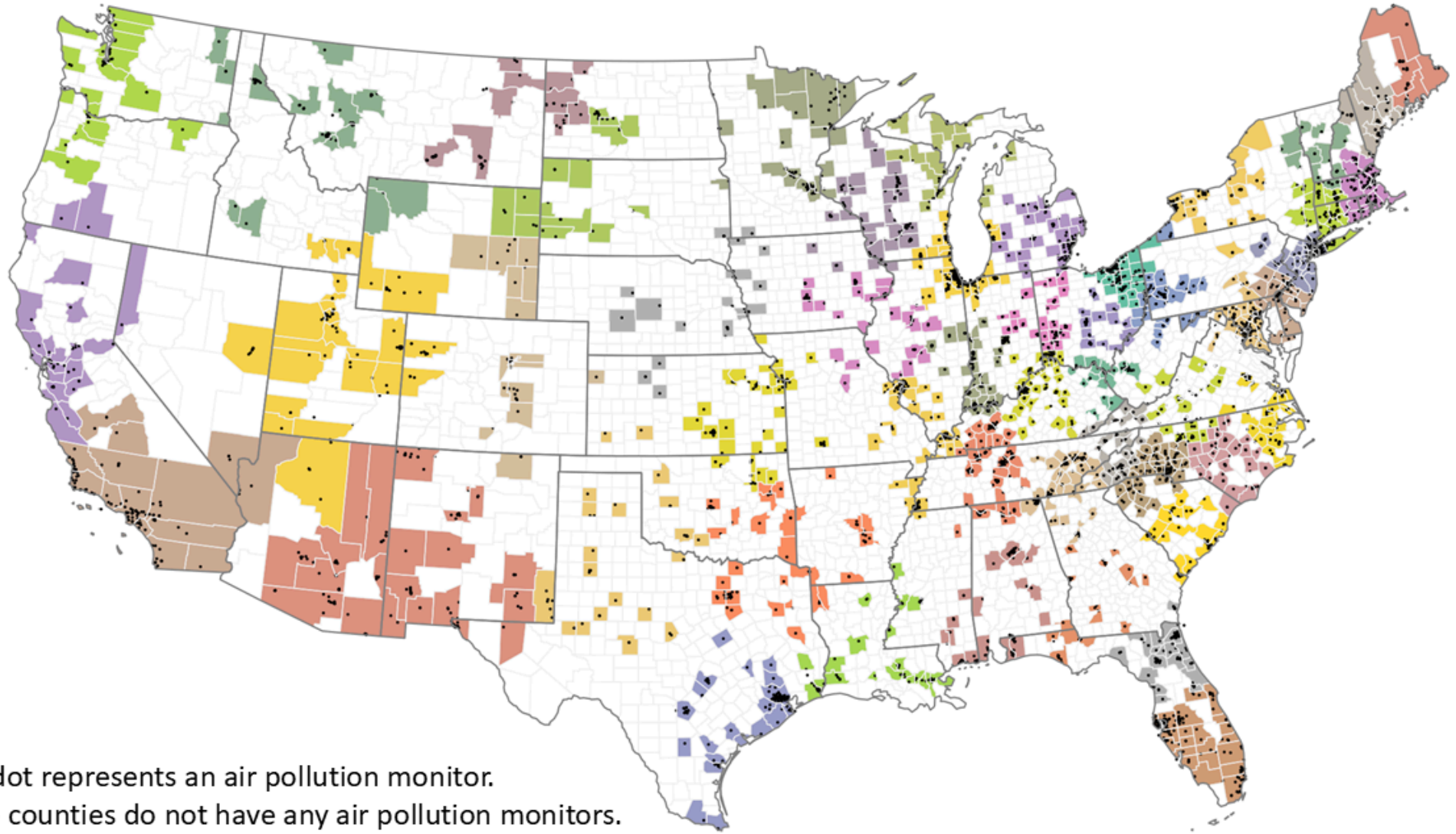
  2. Unsupervised learning: understand relationships between variables
     - X: input variables
     - No output variable

This course focuses on supervised learning

- Regression, lasso, decision trees, neural networks, random forest
  - These are all supervised learning!

- Common types of unsupervised learning:
  - Cluster analysis
  - Principal components analysis (PCA)

Cluster analysis example: air pollution monitors



Each dot represents an air pollution monitor.
White counties do not have any air pollution monitors.
Cluster analysis assigns counties to 50 different groups (denoted by color).

PCA example: measuring productivity

- Illinois Workplace Wellness Study: how do you measure productivity?
- Use 15 different measures of productivity to create a single measure

Productivity measure	Effect on productivity (PCA result)
Sick leave (days/year) [admin]	-0.064
Any sick days in past year [survey]	-0.050
Worked 50+ hours/week [survey]	-0.035
Management priority on health/safety [survey]	0.229
Annual salary (dollars) [admin]	N/A
Very or somewhat satisfied with job [survey]	0.342
Very satisfied with job [survey]	0.320
Job search somewhat/very likely [survey]	-0.307
Job search very likely [survey]	-0.283
Feel very productive at work [survey]	0.209
Annual salary (share of baseline salary) [admin]	0.232
Received promotion [survey]	0.250
Happier at work than last year [survey]	0.306
Job title change [admin]	0.379
Job promotion [admin]	0.386

Source: Jones, Molitor, and Reif (2019, *Quarterly Journal of Economics*)

Supervised learning model

Dependent variable, Y

Independent variables, $X = (X_1, X_2, \ldots, X_p)$

Model:

$$Y = f(X) + \epsilon$$

where the error term, ϵ, is random noise with mean zero: $E[\epsilon] = 0$

- Our goal is to estimate f: the systematic information that X provides about Y
  - f is sometimes referred to as the "conditional expectation function" (CEF)

Types of supervised learning models

- Univariate vs multivariate
  - Univariate: $X = X_1$
  - Multivariate: $X = (X_1, X_2, \ldots, X_p)$ where $p \geq 2$

- Parametric vs non-parametric
  - Parametric models make a lot of assumptions about the form of f (eg, regression)
  - Non-parametric models are more flexible, but require more data (eg, decision trees)

- Linear vs non-linear
  - Linear regression: $Y = \beta X + \epsilon$
  - Nonlinear: decision trees

Notation

- The "true" model, f, is fixed:

$$Y = f(X) + \epsilon$$

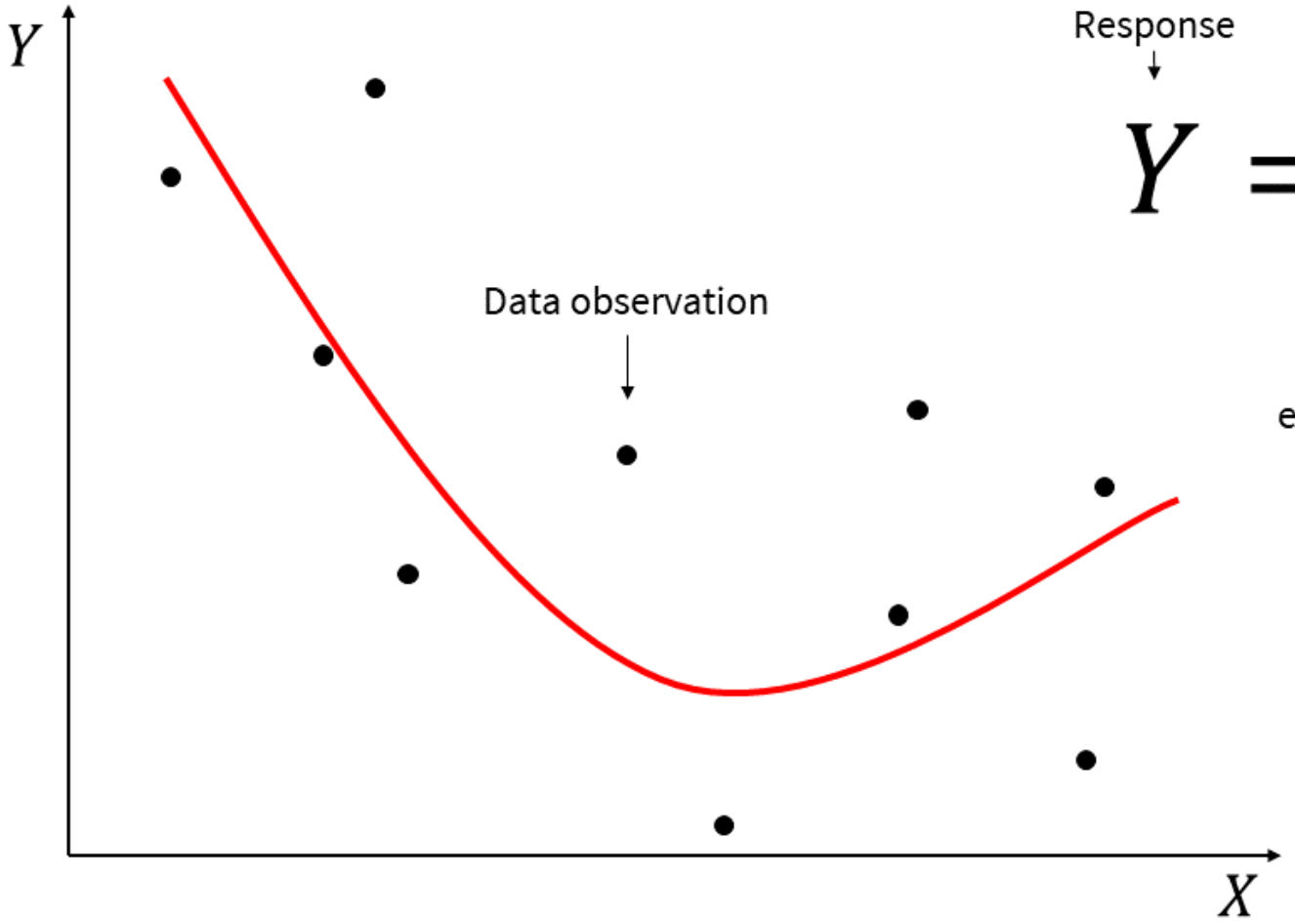
- Our *estimate* of f is denoted as $\hat{f}$ ("f hat")
  - This estimate produces a set of predictions, $\hat{Y} = \hat{f}(X)$
  - Note: $Y \neq \hat{Y}$ (why?)

- In general, $\hat{f} \neq f$
  - We may not be using the most appropriate statistical technique
  - The estimate $\hat{f}$ depends on a dataset that has sampling error $(\epsilon \neq 0)$
  - See lab assignment

Training data vs test data

- Training data: used to estimate the model ($\hat{f}$)
- Test data: used to evaluate the model
- A model that fits the training data well might not fit the test data well

How the world works (omnisicient view)



Response ↓ Predictor(s) ↓ Irreducible error ↓

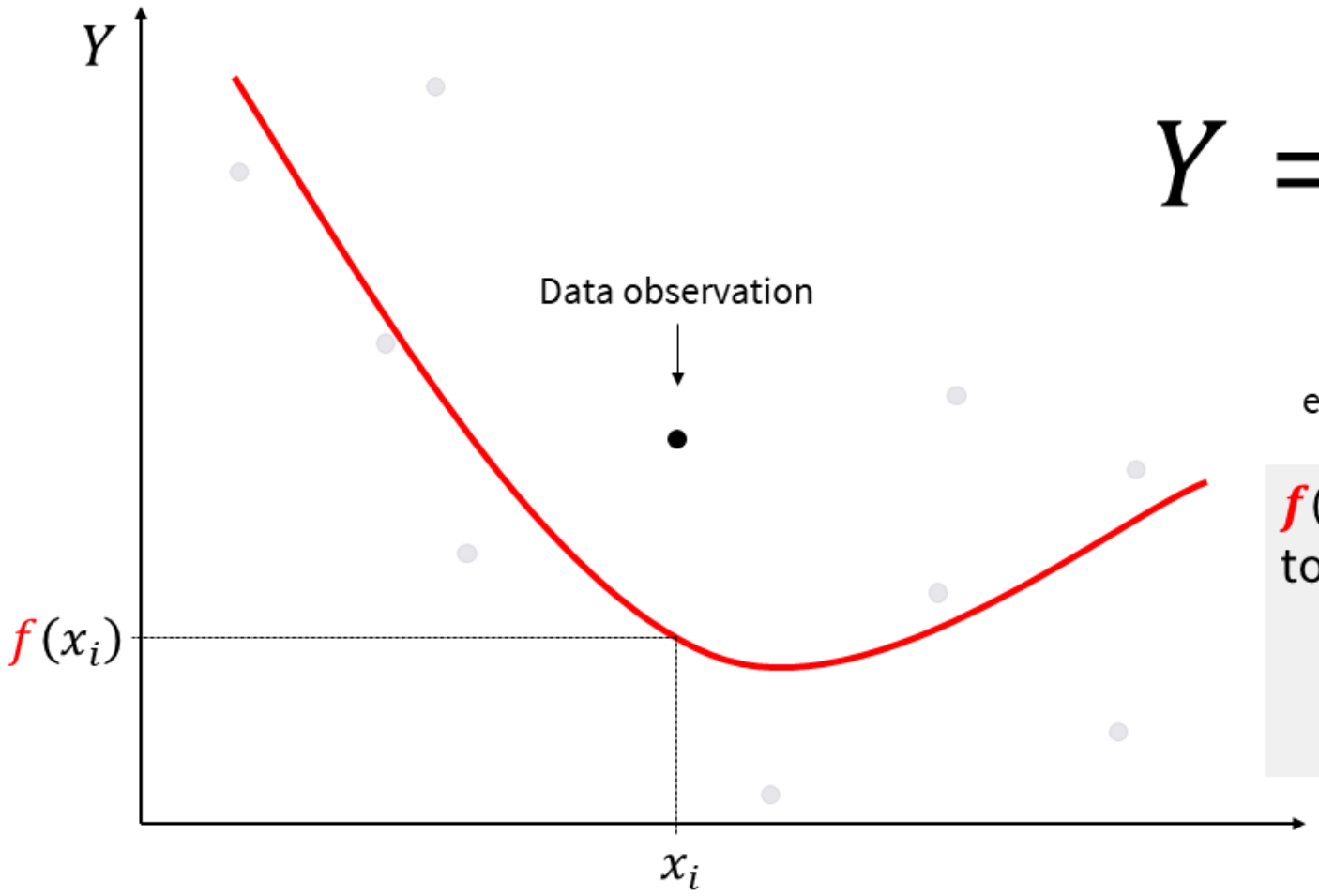
$$Y = f(X) + \epsilon$$

$f = E[Y|X]$

"Conditional expectation function"

$E[\epsilon] = 0$

How the world works (omnisicient view)



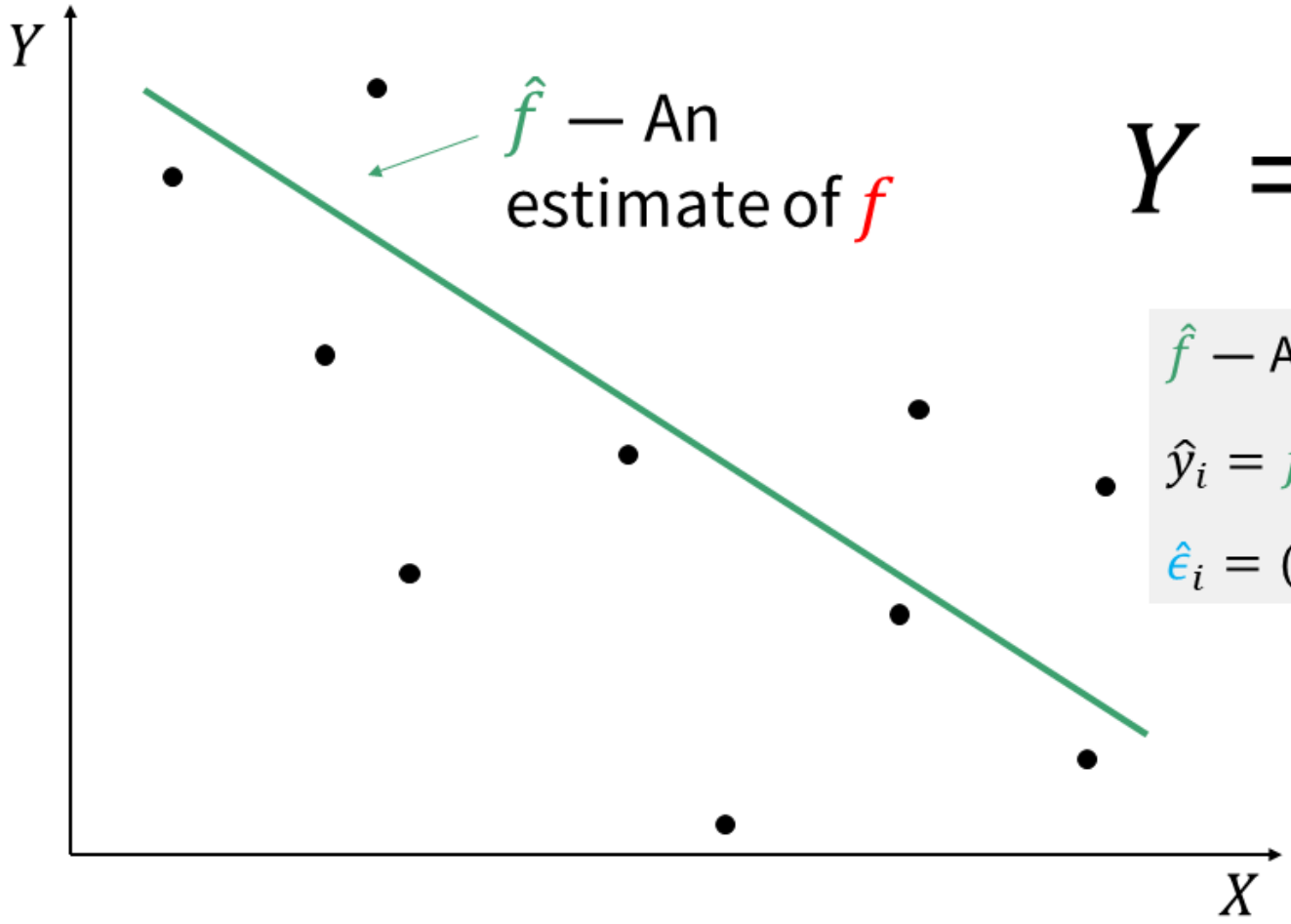
$$Y = f(X) + \epsilon$$

$f = E[Y|X]$ "Conditional expectation function"

$E[\epsilon] = 0$

$f(x_i)$ — The outcome we'd expect to find, on average, when $X = x_i$

How the world works (our view)



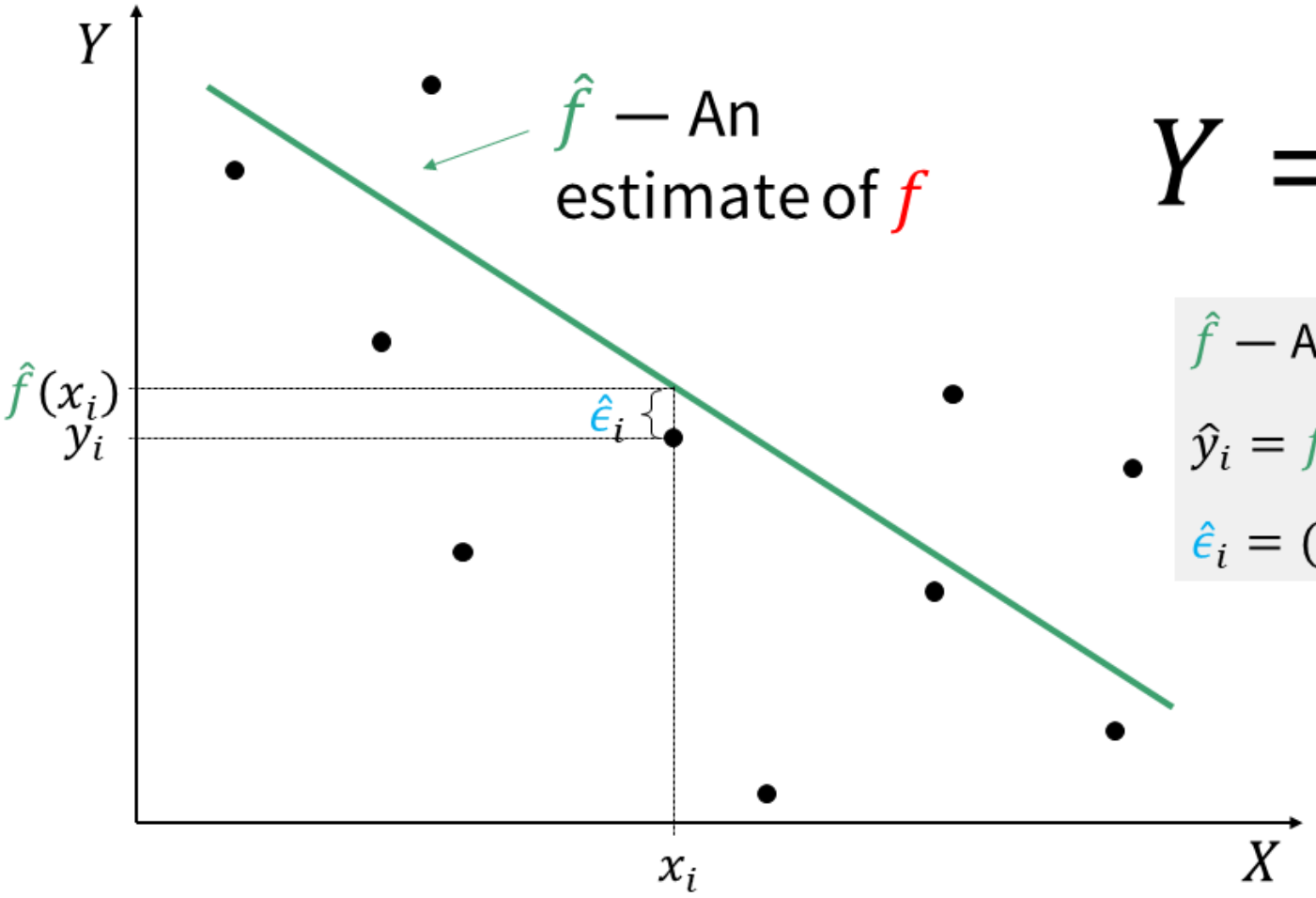
$$Y = f(X) + \epsilon$$

$\hat{f}$ — An estimate of f

$\hat{y}_i = \hat{f}(x_i)$ — The predicted outcome

$\hat{\epsilon}_i = (y_i - \hat{y}_i)$ — Prediction error

How the world works (our view)



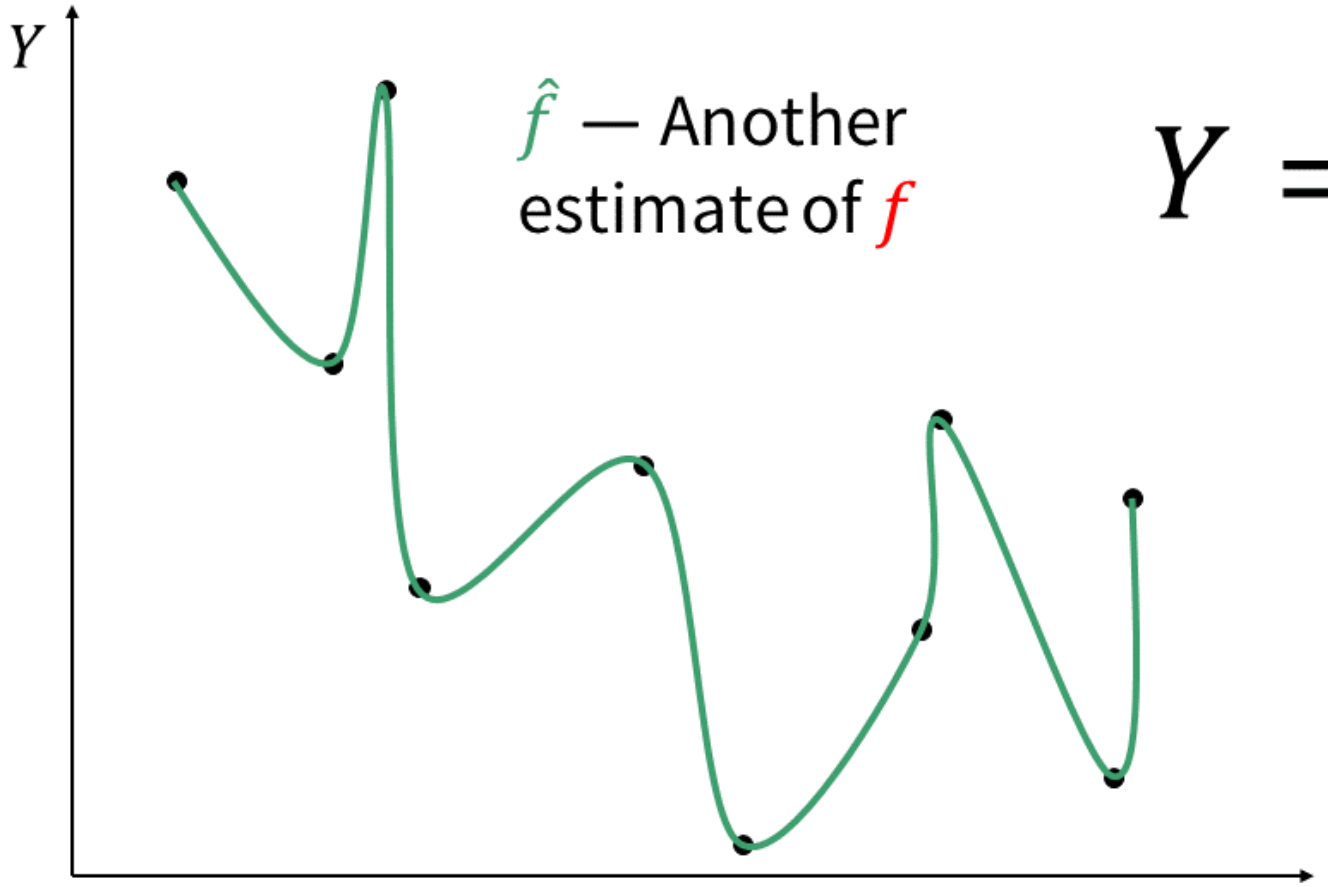
$$Y = f(X) + \epsilon$$

$\hat{f}$ — An estimate of f

$\hat{y}_i = \hat{f}(x_i)$ — The predicted outcome

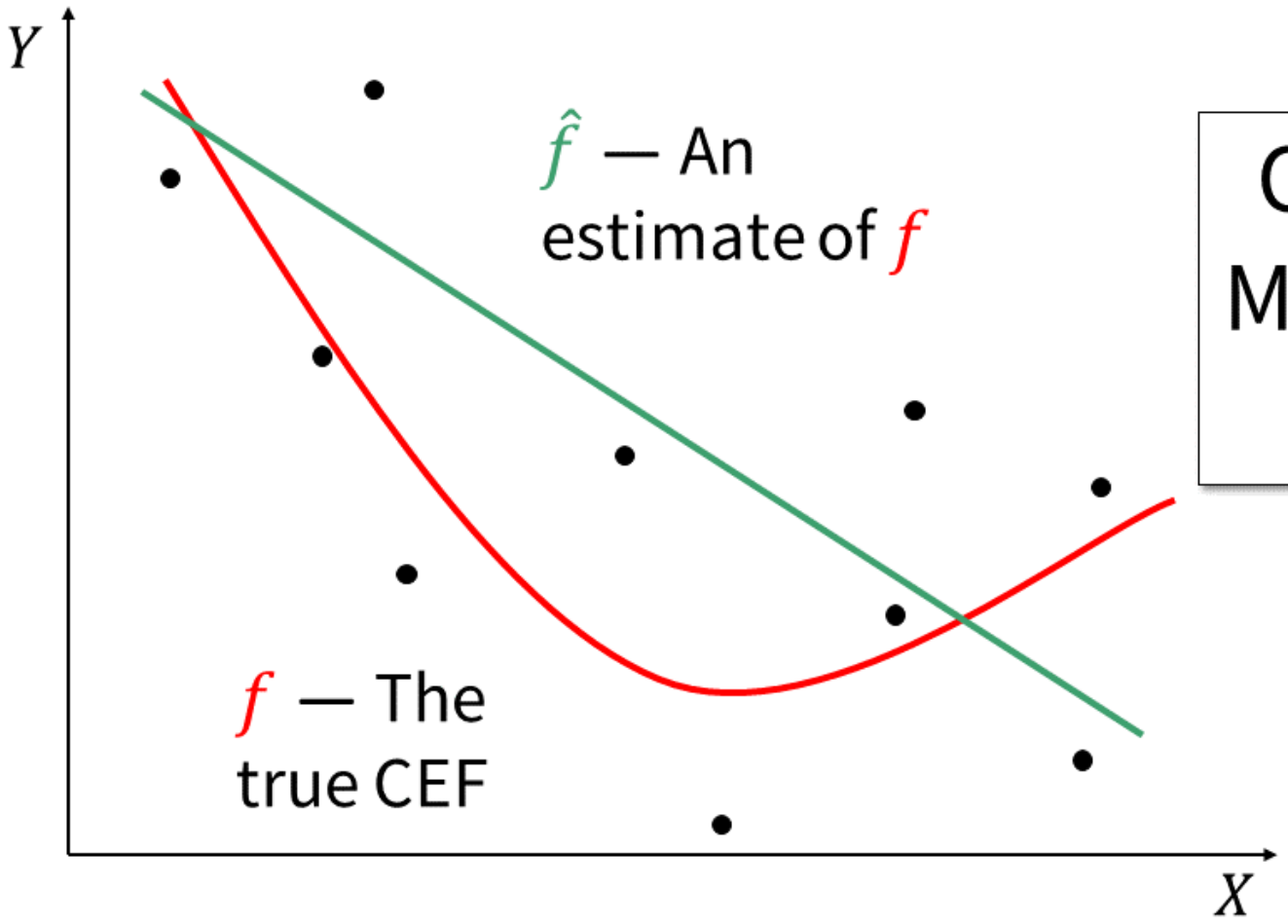
$\hat{\epsilon}_i = (y_i - \hat{y}_i)$ — Prediction error

How the world works (our view)



$$Y = f(X) + \epsilon$$

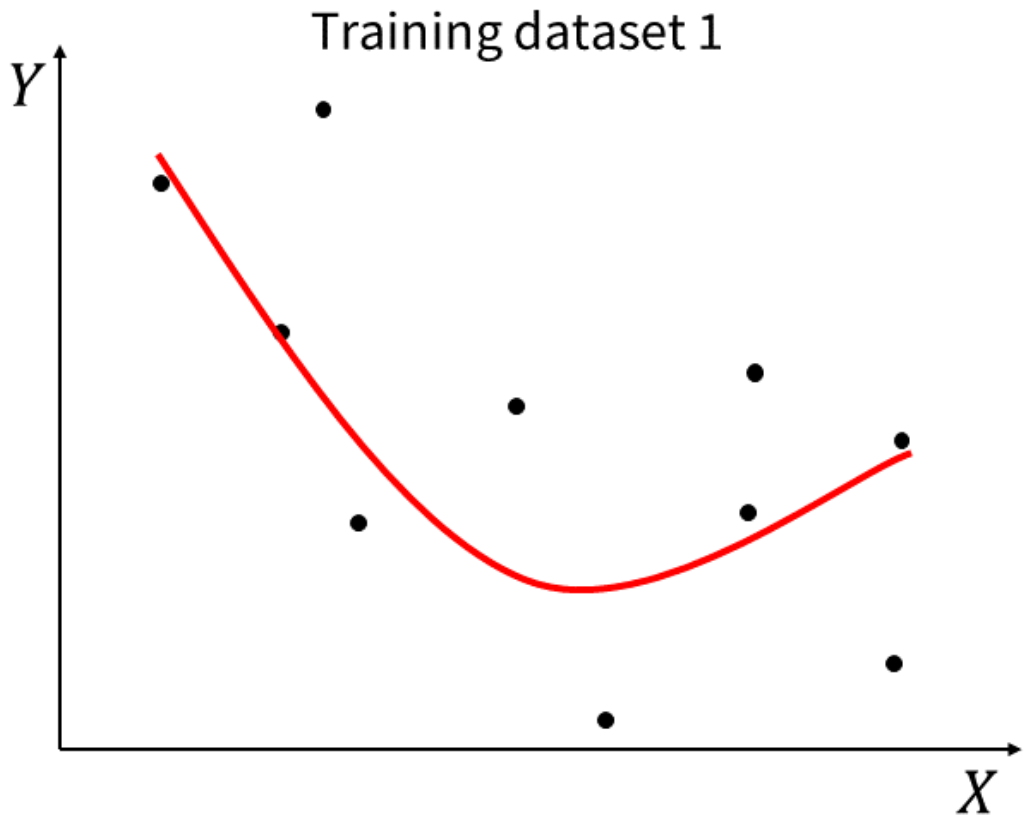
Bias: prediction error due to wrong assumptions

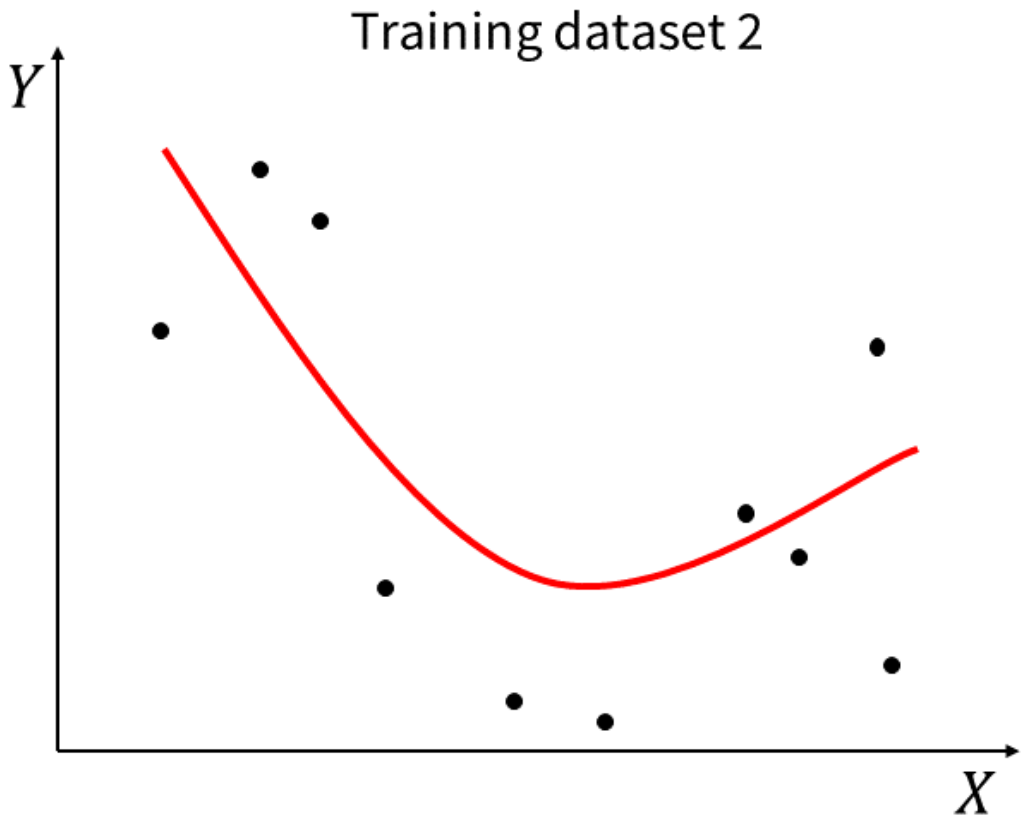


Overly simplistic ML algorithms can lead to **bias**

Variance: the sensitivity of $\hat{f}$ to training data

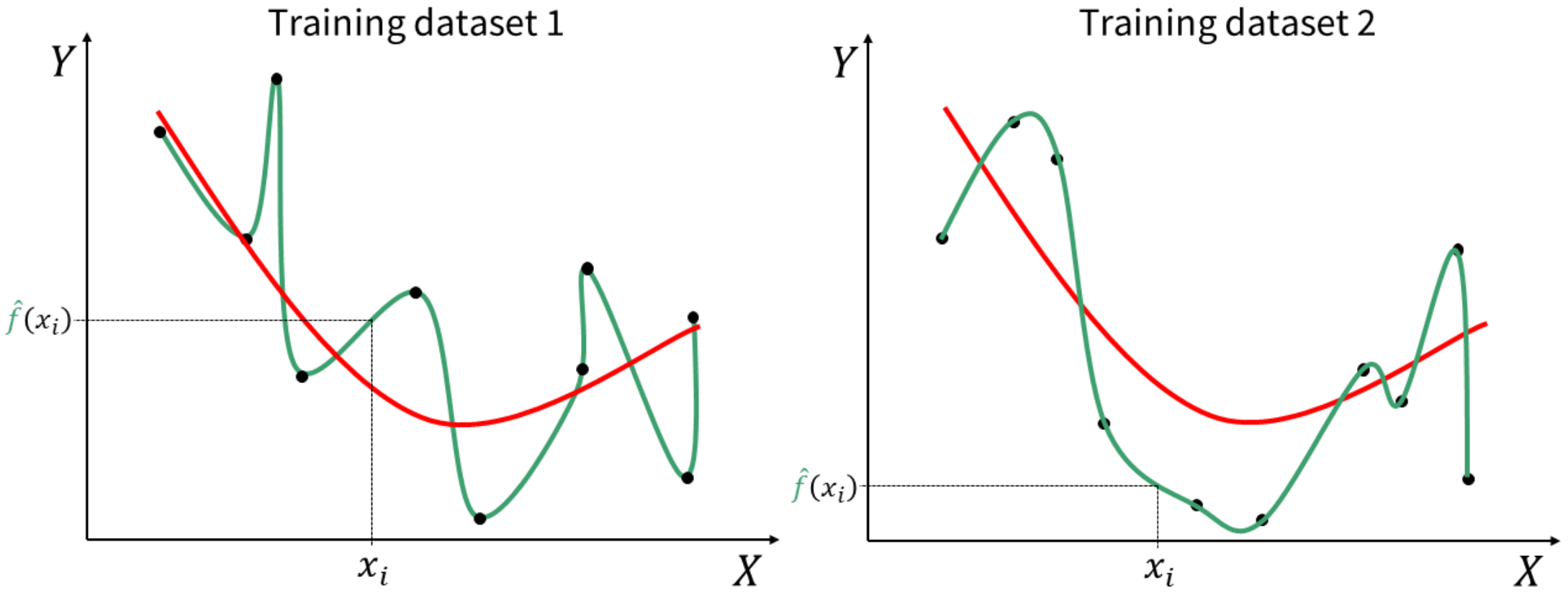
Overly flexible ML algorithms can lead to high **variance**





Variance: the sensitivity of $\hat{f}$ to training data

Overly flexible ML algorithms can lead to high **variance**



Evaluating performance

- Ideally, we have low bias and low variance
- These are not directly observed, however (exception: simulation!)
- Evaluate algorithms using mean-squared error (MSE)

$$MSE = E\left[\hat{\epsilon}^2\right]$$

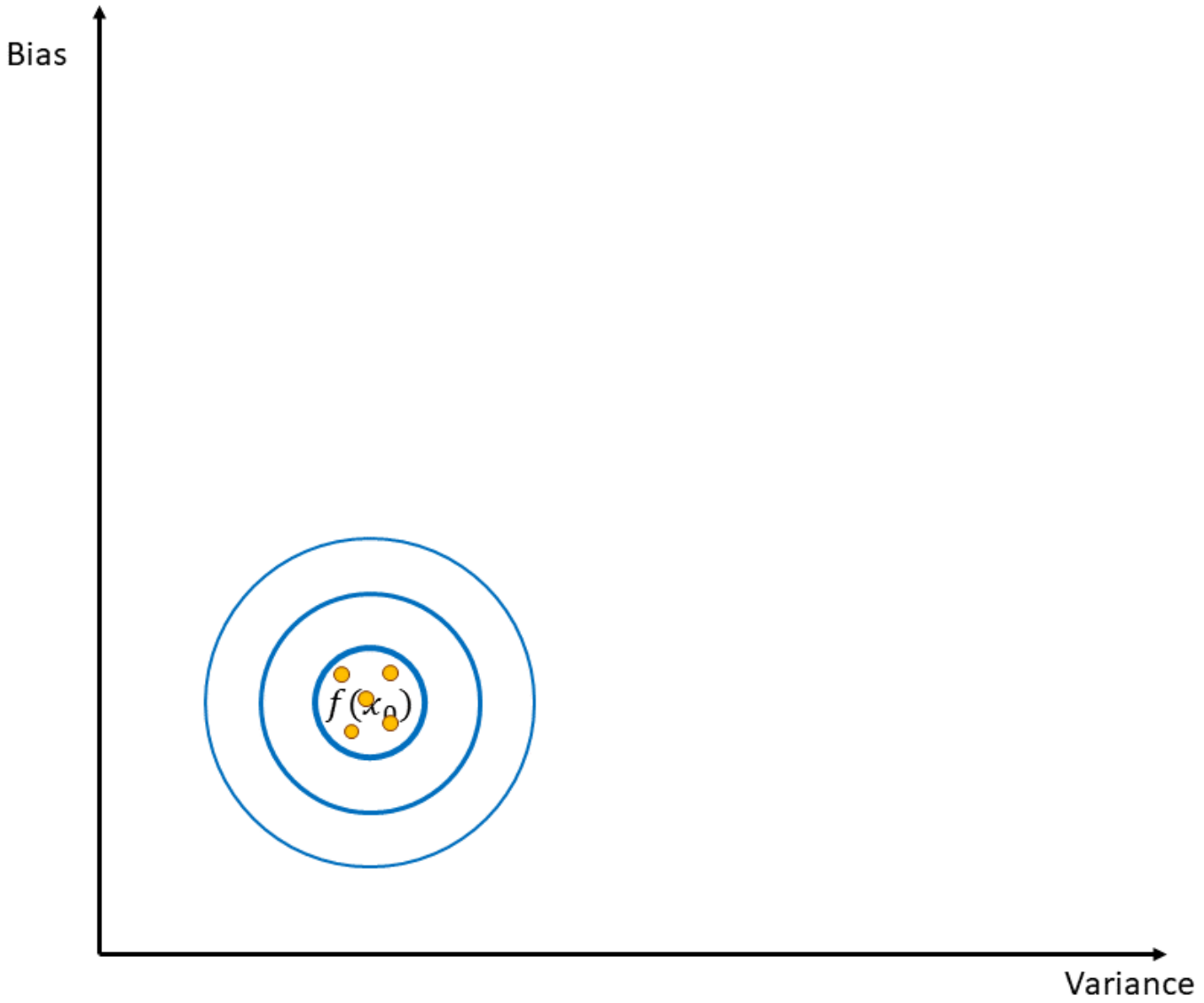
$$= E\left[\left(\underbrace{Y - \hat{Y}}_{\text{prediction error}}\right)^2\right]$$

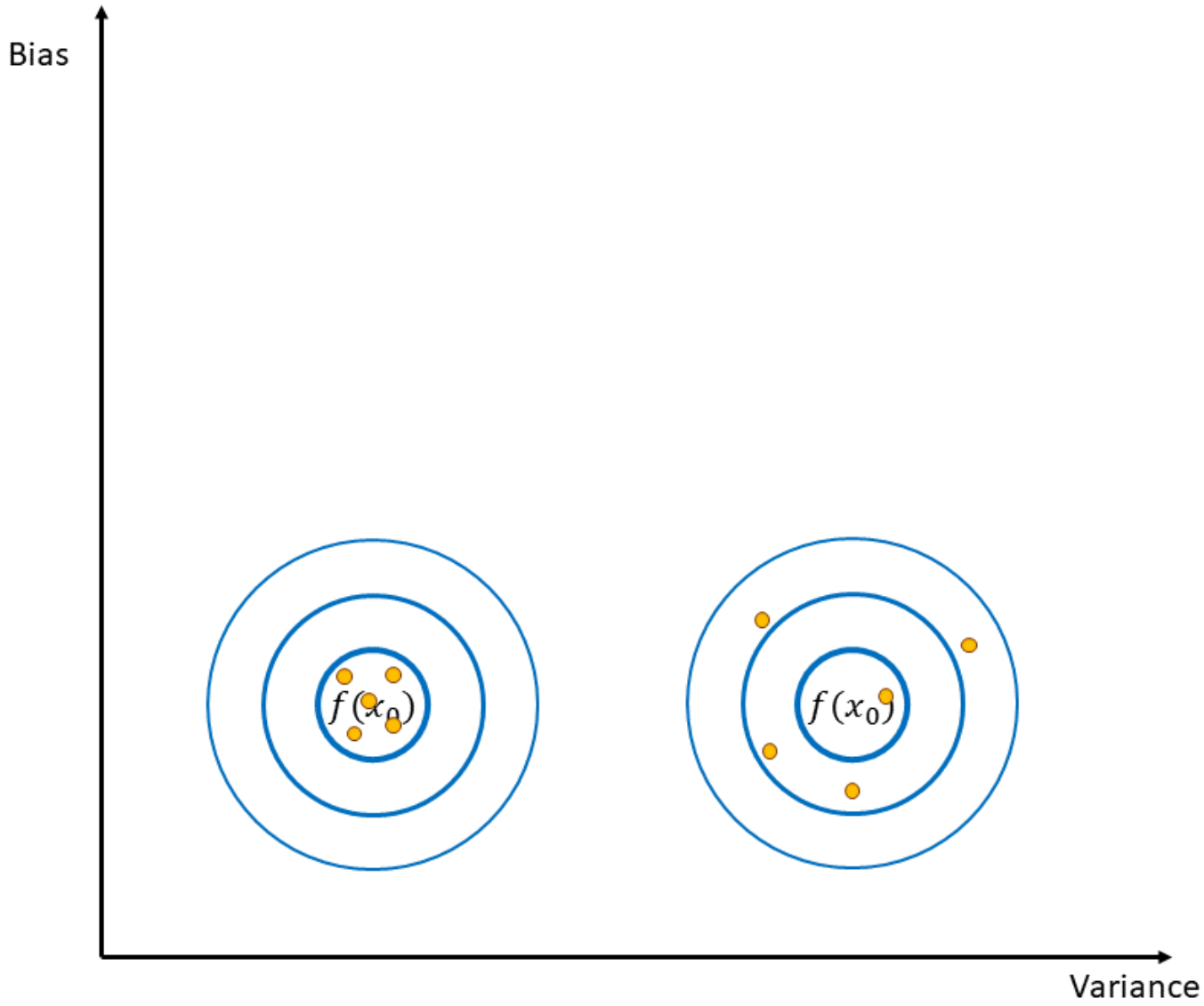
MSE decomposition

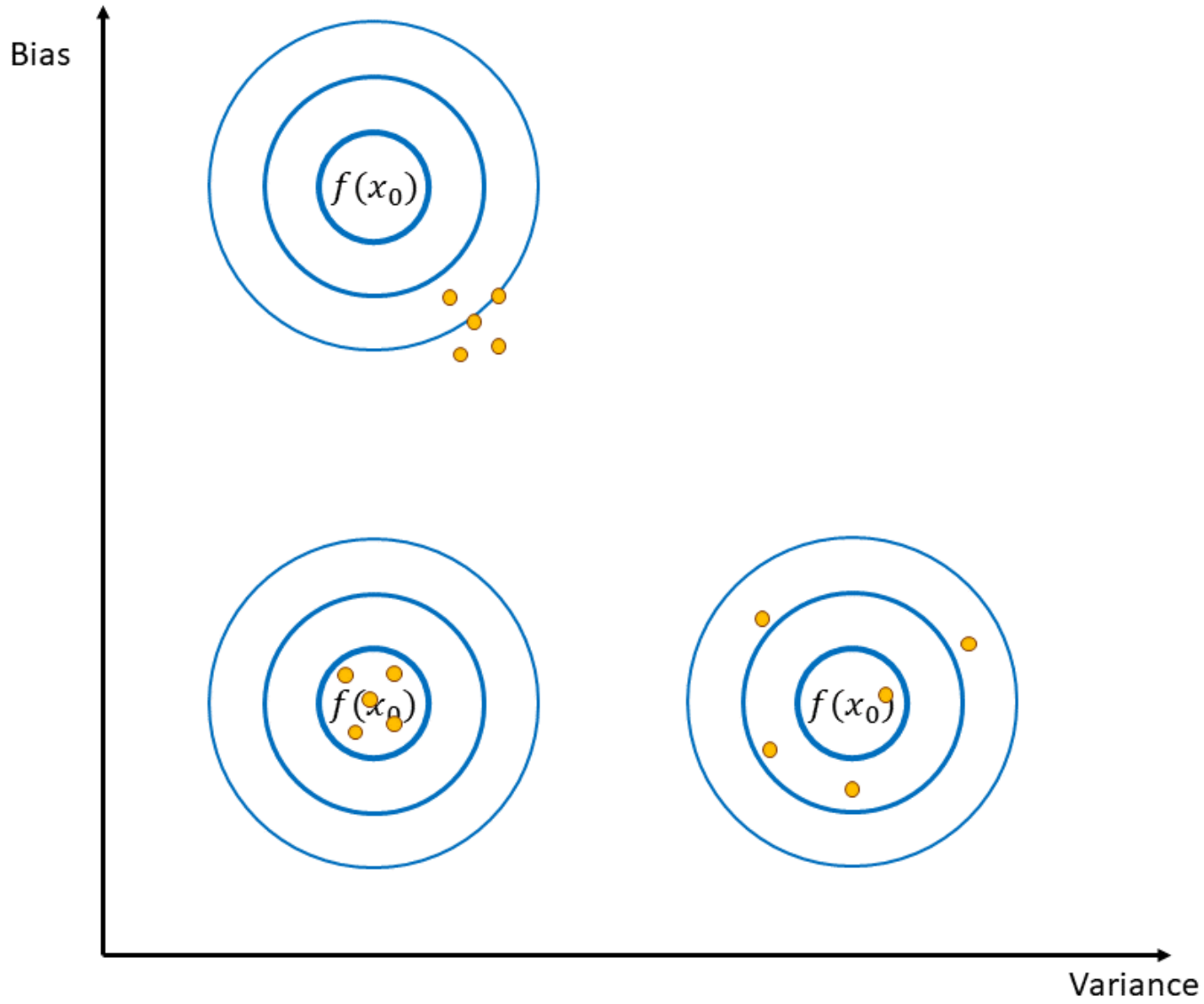
$$\underbrace{MSE}_{\text{Mean-squared error}} = \underbrace{E\left[\epsilon^2\right]}_{\text{Noise}} + \underbrace{\left(f - E\left[\hat{f}\right]\right)^2}_{\text{Bias}^2} + \underbrace{Var\left(\hat{f}\right)}_{\text{Variance}}$$

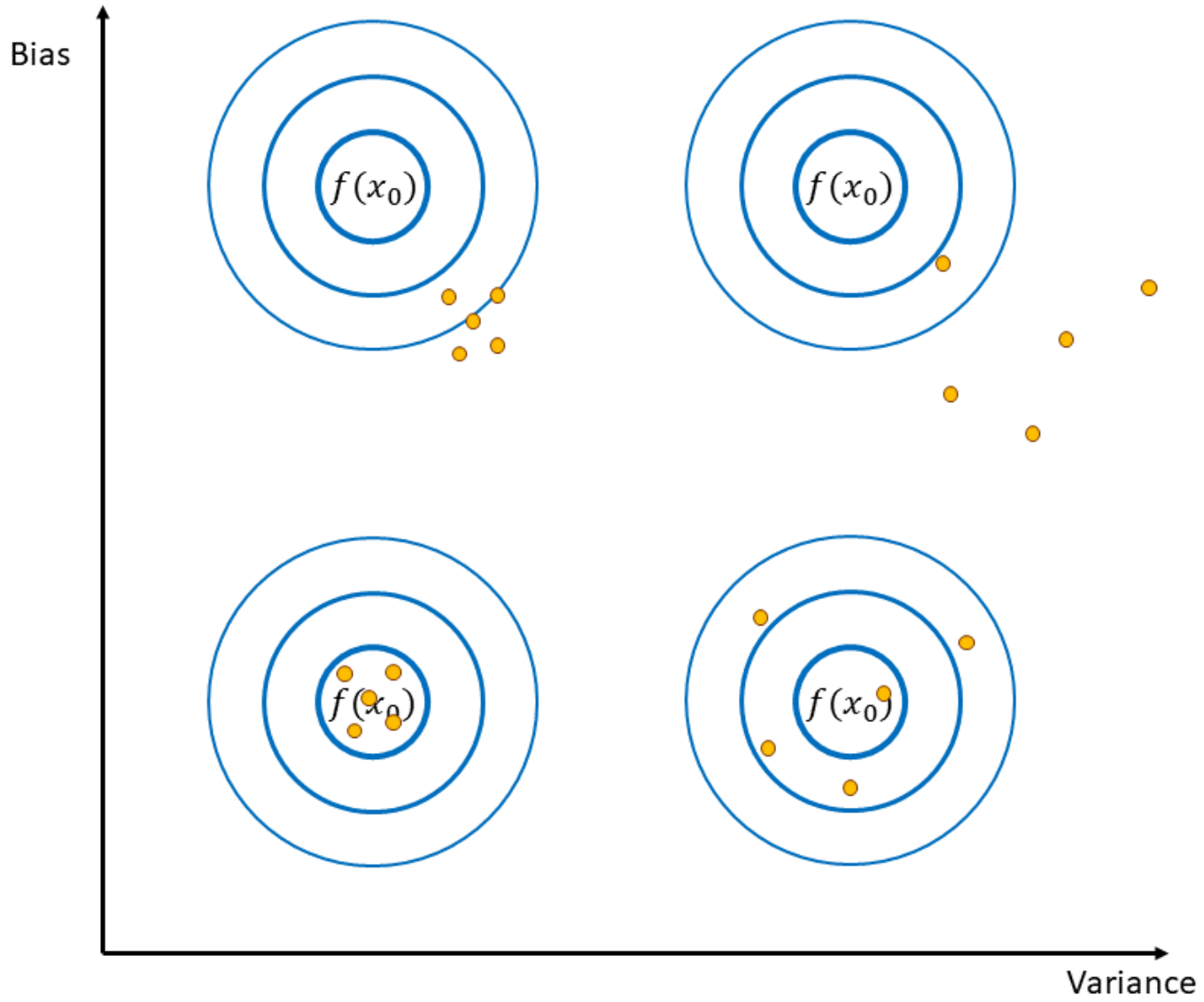
- Noise: Irreducible (at least with the current predictors)
- Bias: How far off the mark the prediction algorithm is, on average
- Variance: How much prediction results vary from different training data samples

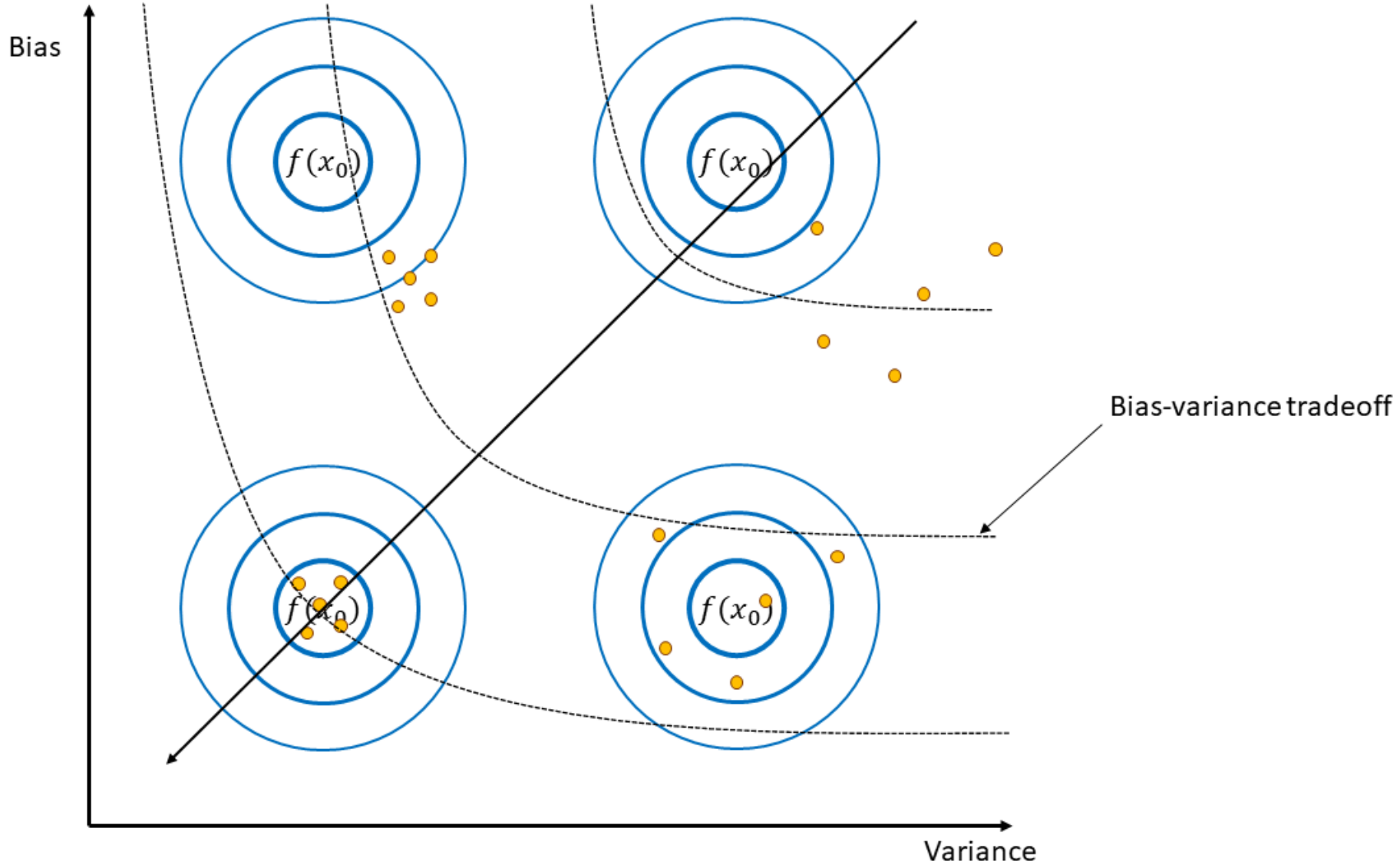
- For a full derivation, see math notes that accompany the lecture slides


Bias
$f(x_0)$
Variance


Bias
$f(x_0)$
$f(x_0)$
Variance


Bias
$f(x_0)$
$f(x_0)$
$f(x_0)$
Variance


Bias
$f(x_0)$
$f(x_0)$
$f(x_0)$
$f(x_0)$
Variance


Bias
$f(x_0)$
$f(x_0)$
$f(x_0)$
$f(x_0)$
Bias-variance tradeoff
Variance

Bias-variance tradeoff: key lessons

1. Simpler ML algorithms tend to have lower variance, higher bias

2. Flexible ML algorithms tend to have higher variance, lower bias

3. The best choice of algorithm will depend on the application

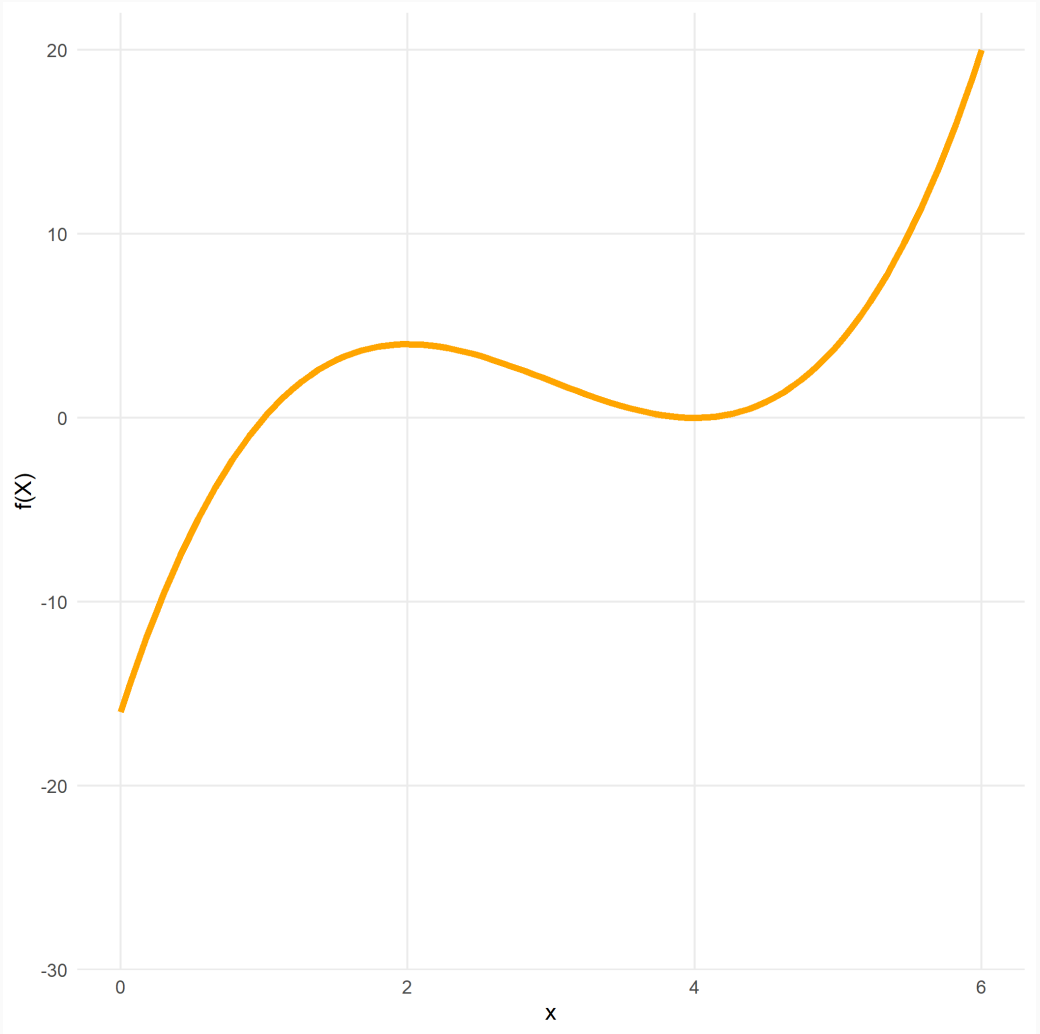
Simulation

Setup:

$$Y = f(X) + \epsilon$$

- X: predictor variable
- f: expected Y given X
- ϵ: noise
- Y: realized outcome

Let f be a cubic function of X

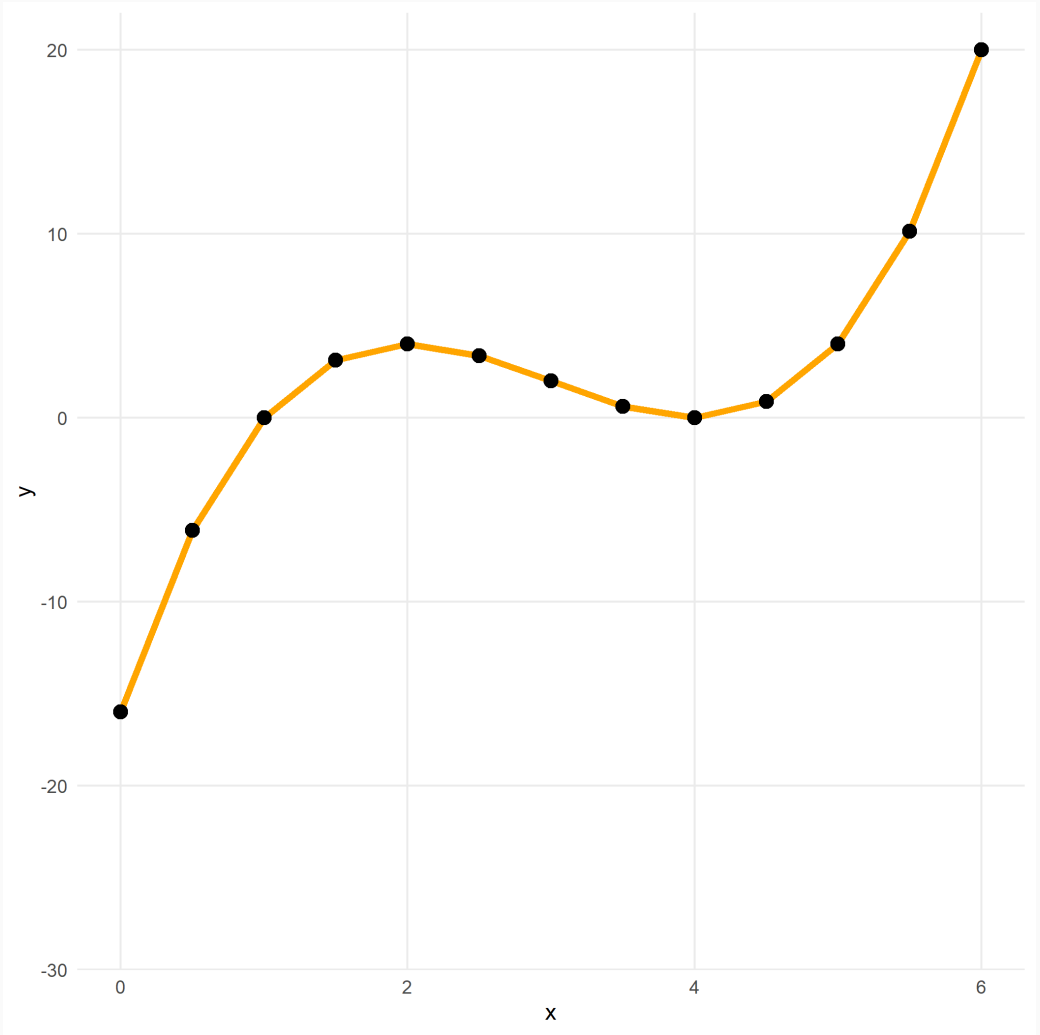


Simulation: sample draw with no noise

Setup:

$$Y = f(X) + \epsilon$$

No noise $\rightarrow$ set $\epsilon = 0$

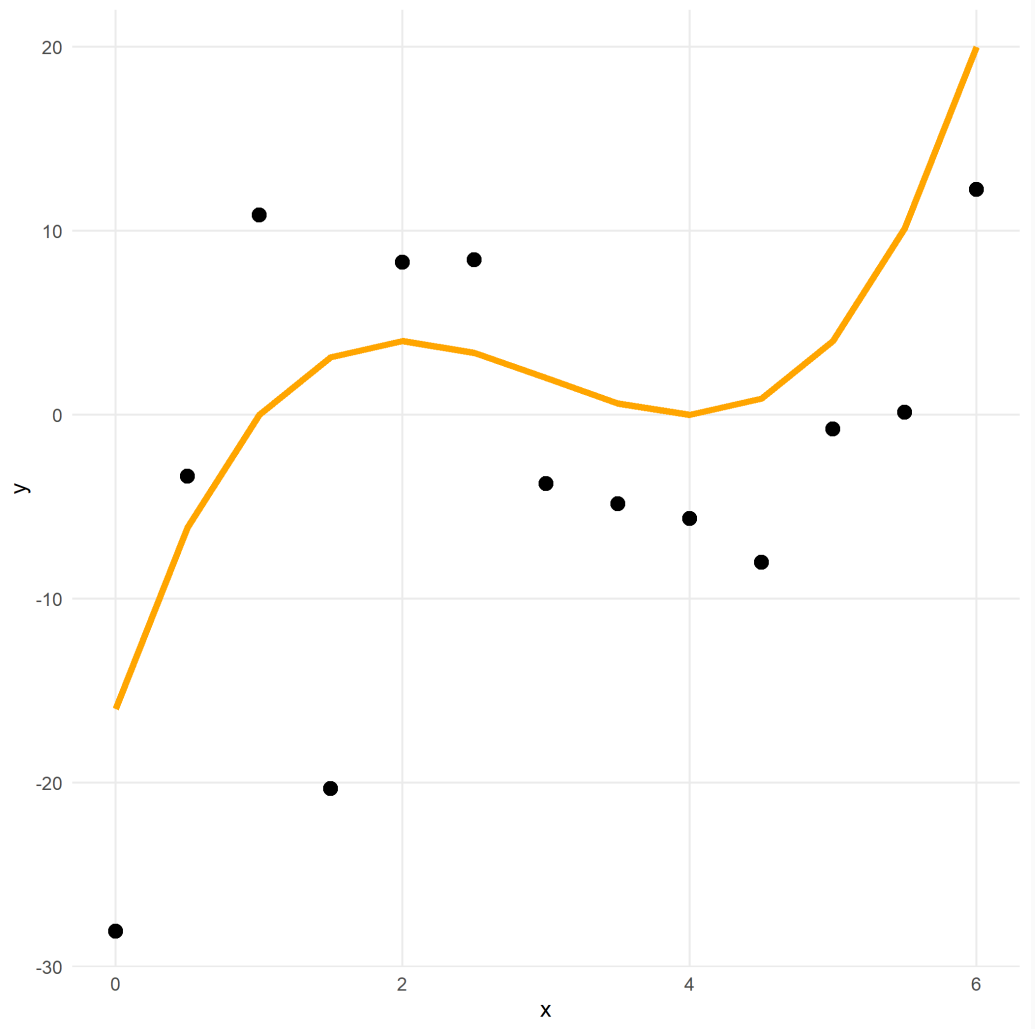


Simulation: sample draw with noise

Setup:

$$Y = f(X) + \epsilon$$

Let $\epsilon \sim N(0, 10)$



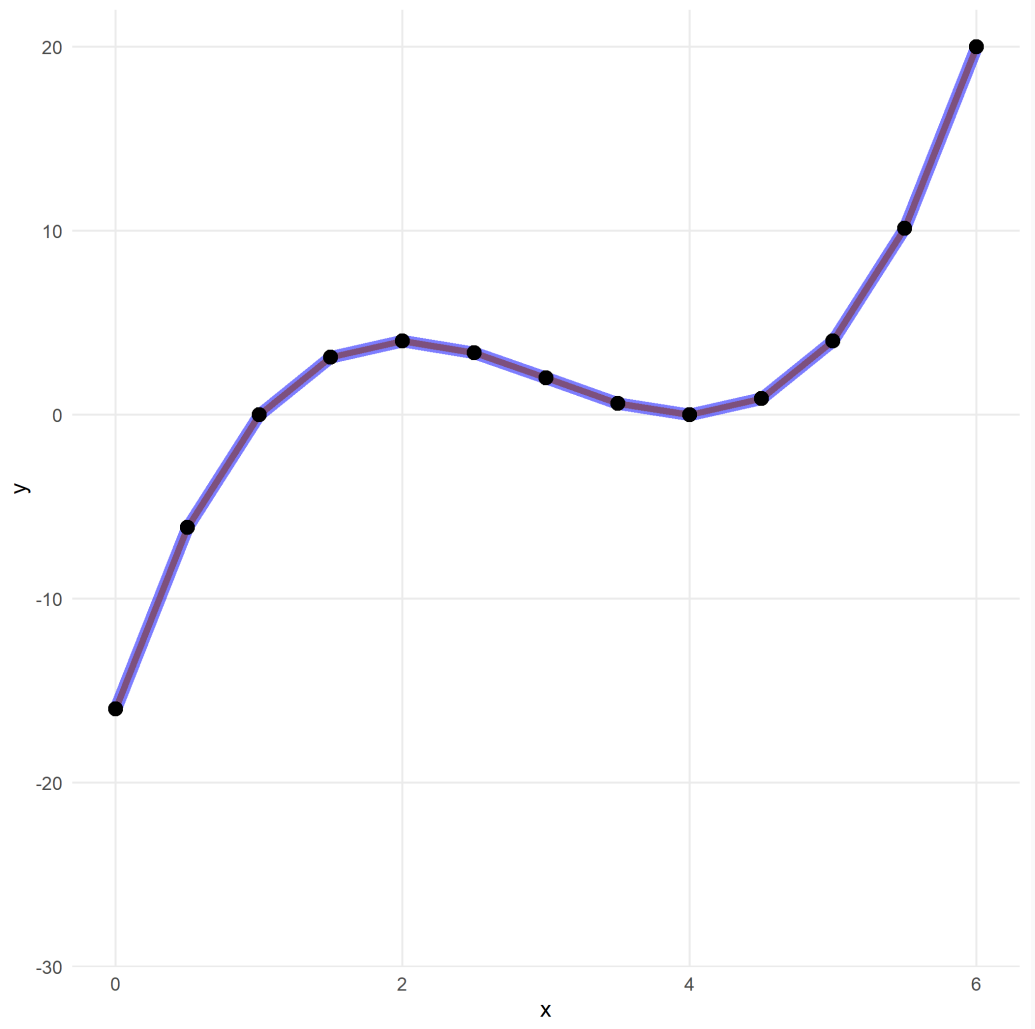
Simulation: machine learning with no noise

Suppose we know that f is a cubic model

Without noise, we fit the data perfectly

The blue line plots $\hat{f}$, the estimate of f obtained from one training sample

Our estimate equals the true f



Simulation: machine learning under noise

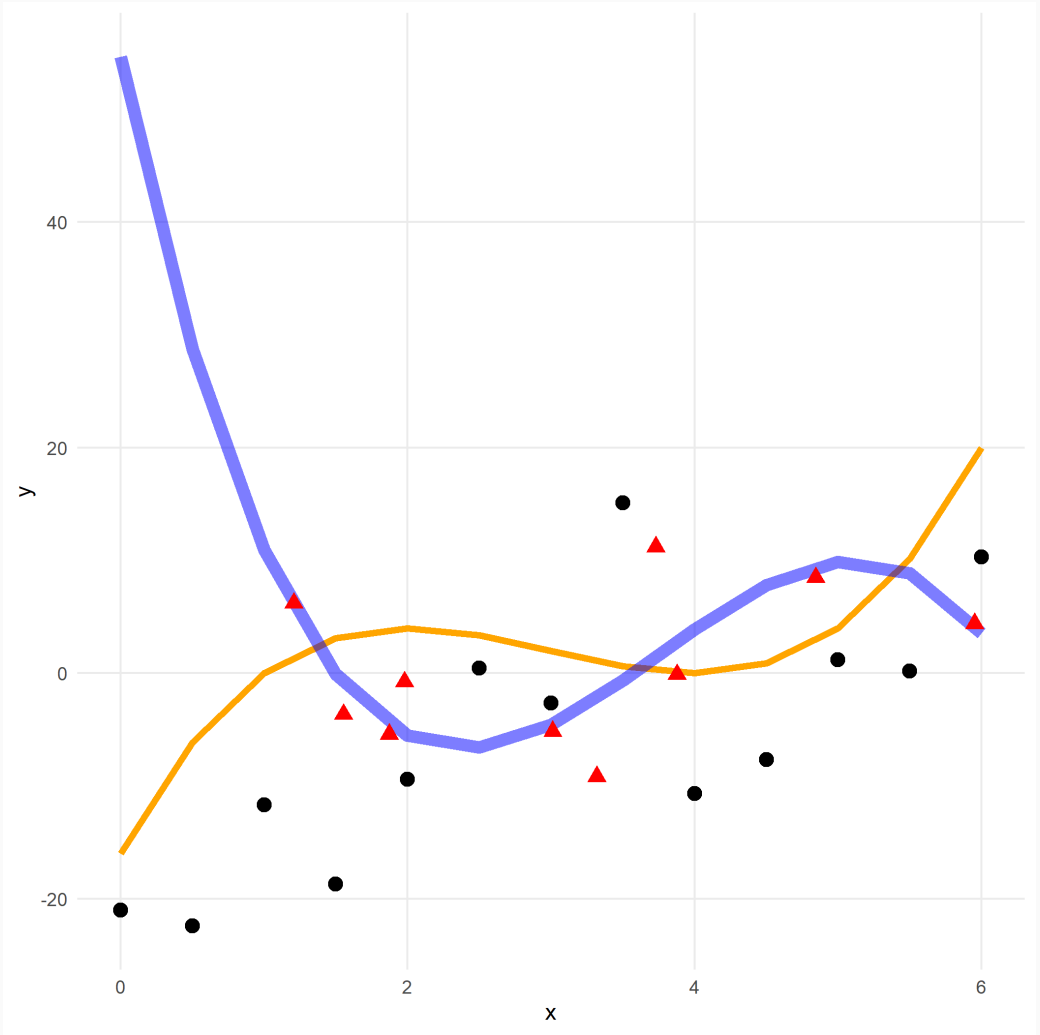
Training data: red triangles

Test data: black dots

When noise is present, $\hat{f} \neq f$

What if we repeatedly drew data samples?

- How much would estimates of f vary?
- How would they look on average?

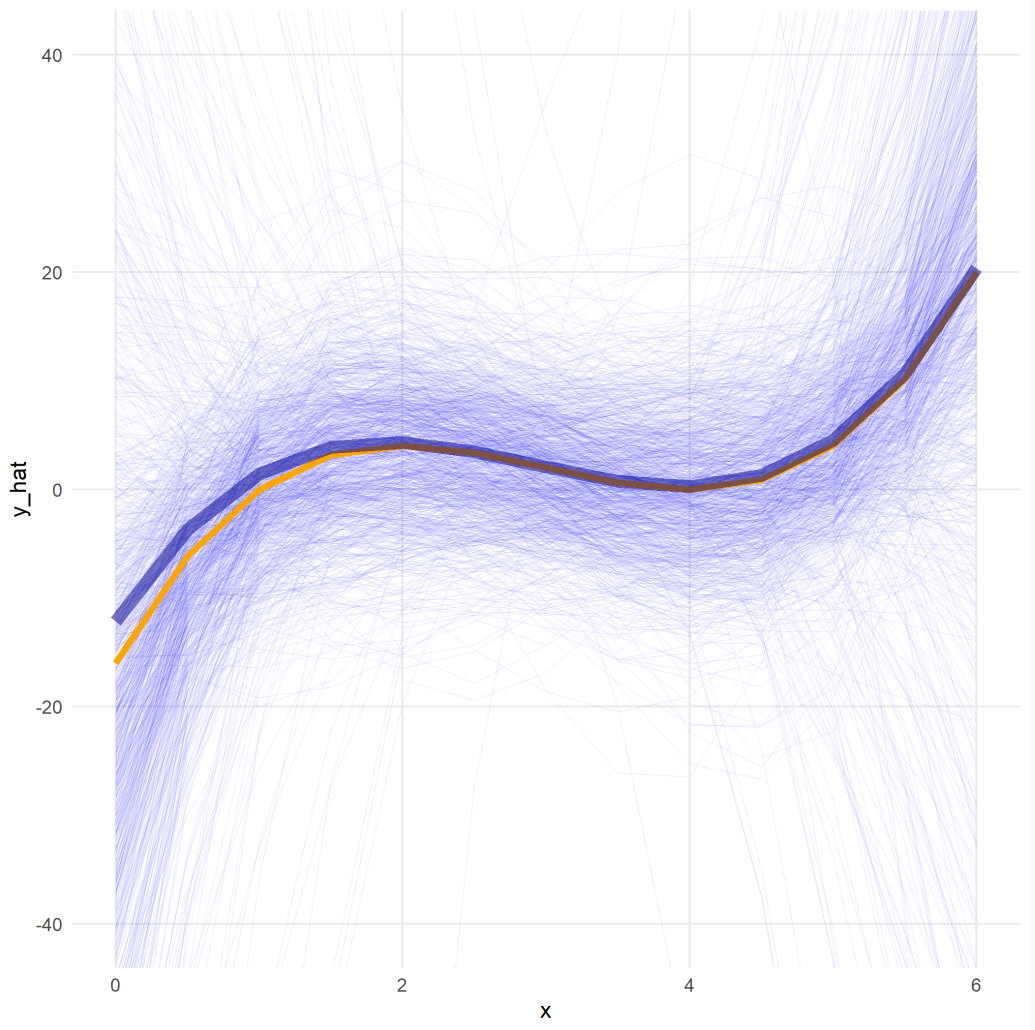


Simulation: assume cubic model

Each thin blue line plots the estimate of f from a separate data sample

The thick blue line plots the average of all the estimates of f

What is the bias?

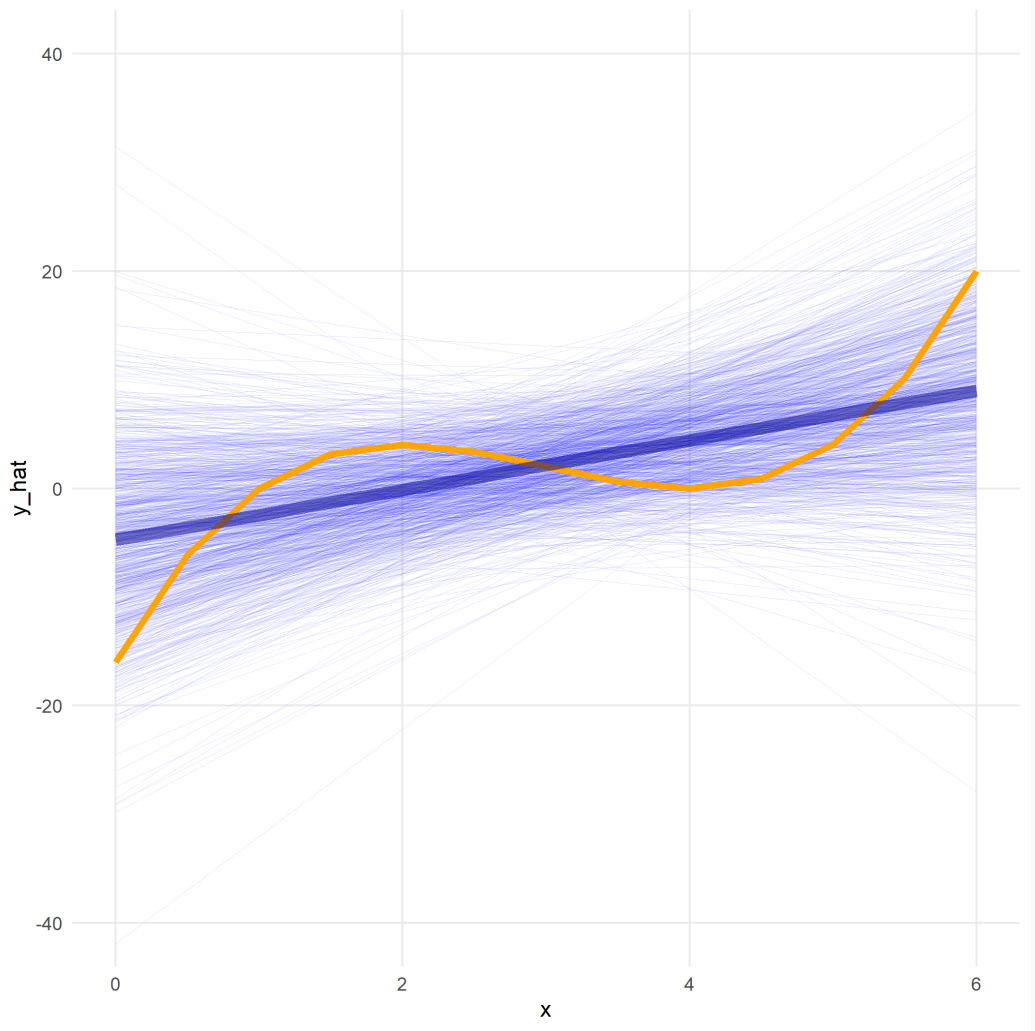


Simulation: assume linear model

Each thin blue line plots the estimate of f from a separate data sample

The thick blue line plots the average of all the estimates of f

How does bias/variance compare to cubic model?



Summary

- Machine learning tries to model systematic relationships between variables
- Predictions suffer from both bias and variance
- Good algorithms have low mean-squared error
- Next topic: linear regression
- Lab-06 is due Sunday at 11:59pm
  - Assignment instructions: Canvas > Weekly Modules
- Make sure to stop your instance when you are done working