Introduction to Machine Learning – Evaluating Models

Dr. Ab Mosca (they/them)

Slides based off slides courtesy of Jordan Crouser (https://jcrouser.github.io/)

Plan for Today

- Evaluating Supervised Machine Learning:
  - Regression
  - Classification
  - Bias-variance trade off
- GitHub Classroom

Recap

Supervised Learning

Find the best function (*f*) for $Y = f(X) + \epsilon$

- Y is the output
- X is the input

- Our data contains ground truth
  - i.e. for the values of X in our data, we know Y

Unsupervised Learning

- Learn patterns from unlabeled data
- Our data does not contain ground truth
  - i.e. we do not know if there are distinct groups in our data

Warm up

Supervised Learning

- Find the best function (f) for $Y = f(X) + \epsilon$
  - Y is the output
  - X is the input
- Our data contains ground truth
  - i.e. for the values of X in our data, we know Y

Unsupervised Learning

- Learn patterns from unlabeled data
- Our data does not contain ground truth
  - i.e. we do not know if there are distinct groups in our data

Practice: Come up with an example of an unsupervised machine learning problem and an example of a supervised machine learning problem.

One model to rule them all…?

Question: why not just teach you the **best** method first?

Answer: there isn't one

- No single method dominates
- One method may prove useful in answering some questions on a given data set
- On a related (not identical) dataset or question, another might prevail

Measuring “quality of fit” for regression models

- *Question we often ask*: how **good** is my model?
- *What we usually mean*: how well do my model’s predictions **actually match** the observations?

How do we choose the **right approach**?

Mean squared error

True response
for the i^{th} observation

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2$$

We take the average
over all observations

Prediction our model gives
for the i^{th} observation

Mean squared error

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2$$

True response for the i^{th} observation

We take the average over all observations

Prediction our model gives for the i^{th} observation

Student	Grade	$f(X)$
Ab	83	78
Kaden	84	85
Kylee	95	65

$n = 3$

$$MSE = \frac{1}{3}\sum_{i=1}^{3}(y_i - \widehat{y_i})^2$$

$$= \frac{1}{3}\left((y_1 - \widehat{y_1})^2 + (y_2 - \widehat{y_2})^2 + (y_3 - \widehat{y_3})^2\right)$$

$$= \frac{1}{3}\left((83 - 78)^2 + (83 - 85)^2 + (95 - 65)^2\right)$$

$$= 309.67$$

“Training” MSE

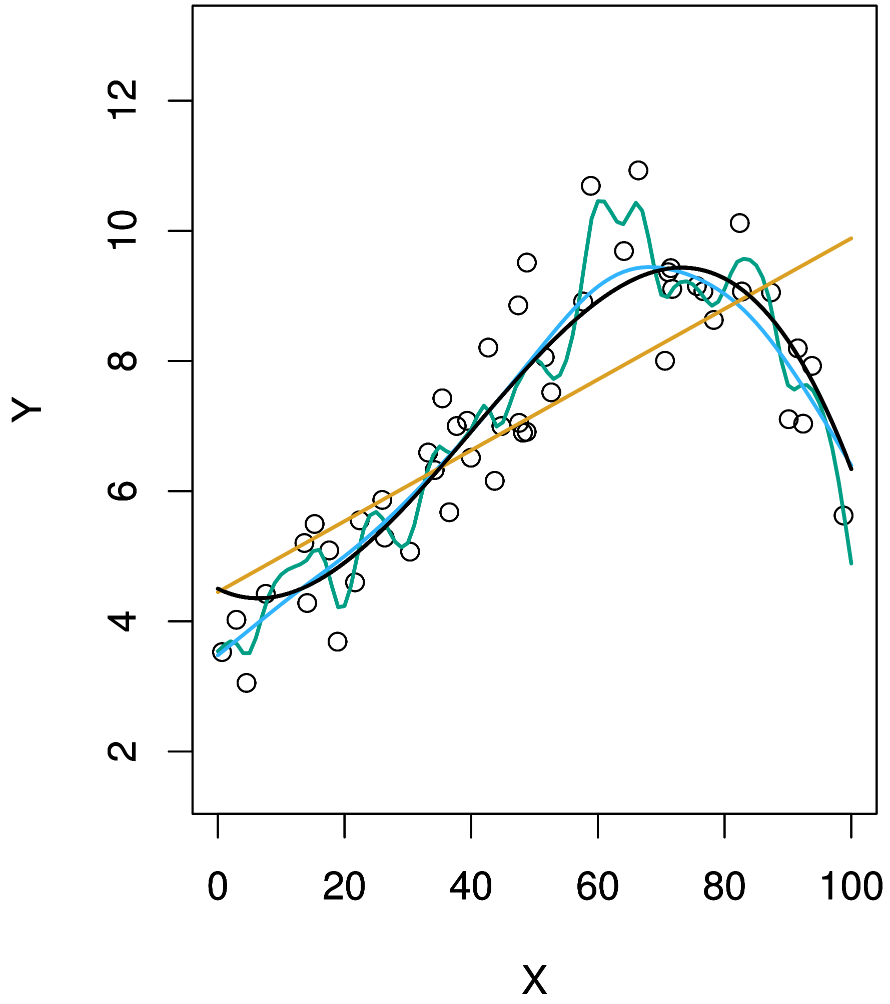
- This version of MSE is computed using the **training data** that was used to fit the model

- **Reality check**: is this what we care about?

Test MSE

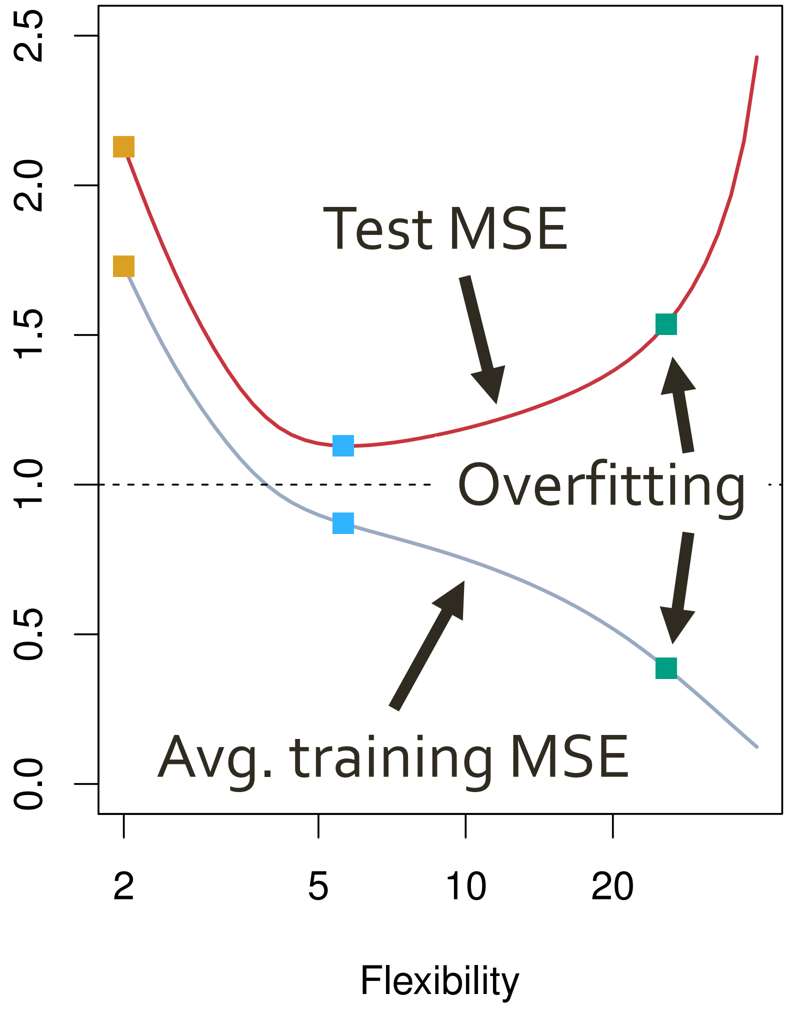
- **Better plan**: see how well the model does on observations we *didn't* train on
- Given some never-before-seen examples, we can just calculate the MSE on those using the same method
- What if we don't have any new observations to test?
  - Can we just use the training MSE?
  - Why or why not?

Example



Training vs. test MSE

- As flexibility ↑:
  - monotone ↓ in training MSE
  - U-shape in the test MSE
- **Fun fact**: occurs regardless of data or statistical method
- This is called **overfitting**



Training vs. test MSE

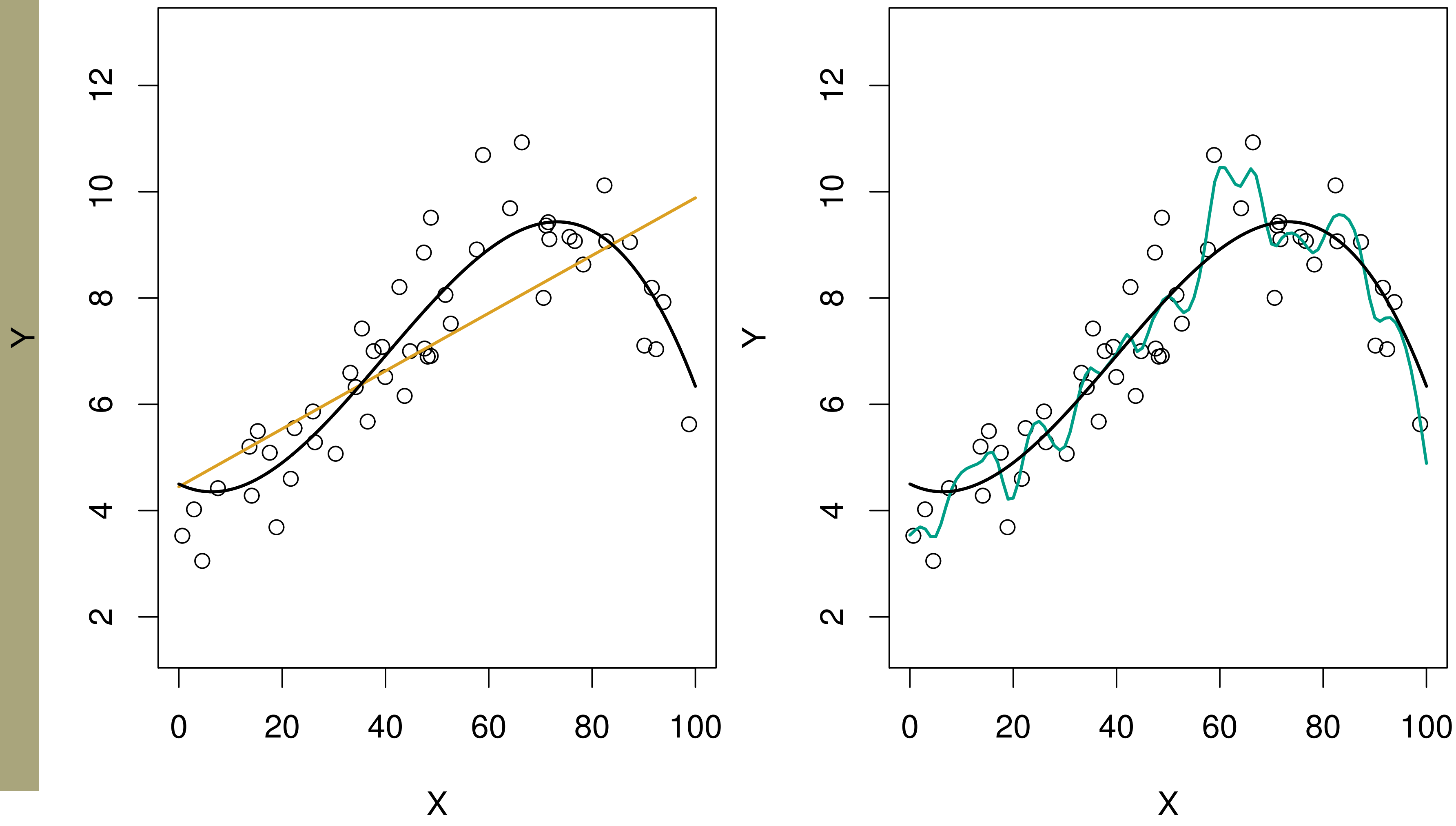
Question: why does this happen?

Trade-off between bias and variance

- The U-shaped curve in the Test MSE is the result of two competing properties: *bias* and *variance*
- **Variance**: the amount the model would change if we had different training data
- **Bias:** the error introduced by approximating a complex phenomenon using a simple model

Relationship between bias and variance

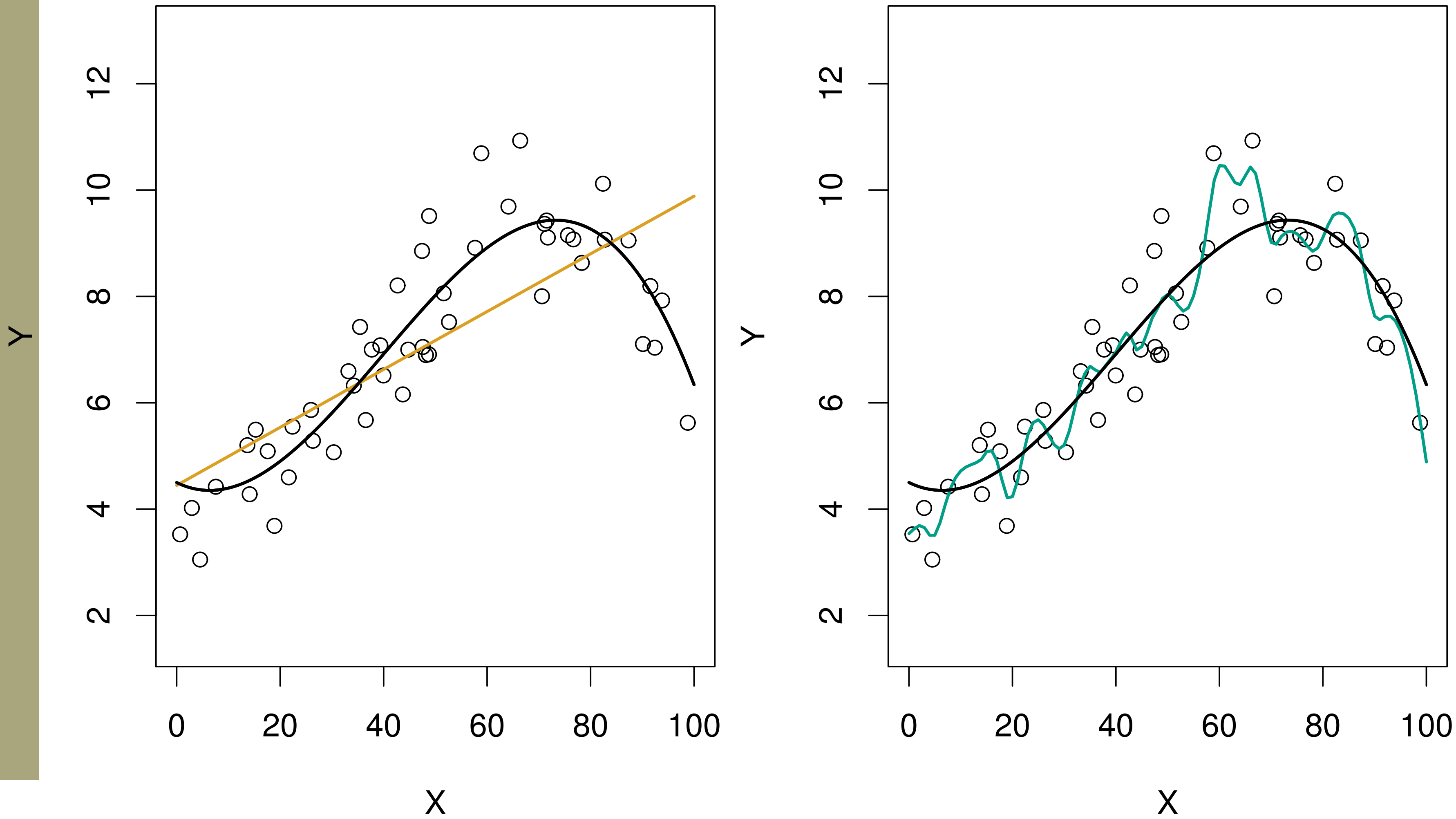
- In general, more flexible methods have **higher variance**



(**Variance**: the amount the model would change if we had different training data)

Relationship between bias and variance

- In general, more flexible methods have **lower bias**



(**Bias:** the error introduced by approximating a complex phenomenon using a simple model)

Trade-off between bias and variance

- Expected test MSE can be decomposed into three terms:

$$E\left(y_0 - \hat{f}(x_0)\right)^2 = Var\left(\hat{f}(x_0)\right) + \left[Bias\left(\hat{f}(x_0)\right)\right]^2 + Var(\varepsilon)$$

The variance of our model
on the test value

The bias of our model
on the test value

The variance
of the error terms

Balancing bias and variance

- It's easy to build a model with

  low variance but **high bias** (how?)

- Just as easy to build one with

  low bias but **high variance** (how?)

- The challenge: finding a method for which both the variance and the squared bias are low

- This trade-off is one of the most important recurring themes in this course

(**Variance**: the amount the model would change if we had different training data
Bias: the error introduced by approximating a complex phenomenon using a simple model)

What about classification?

- So far: how to evaluate a **regression** model
- Bias-variance trade-off also present in **classification**
- Need a way to deal with **qualitative responses**

What are some options?

Training error rate

- **Common approach**: measure the proportion of the times our model incorrectly classifies a training data point

and take the average

$$\frac{1}{n}\sum_{i=1}^{n} I\left(y_i \neq \hat{y}_i\right)$$

tally up all the times

where the model's classification was **different** from the true class

Training error rate

- **Common approach**: measure the proportion of the times our model incorrectly classifies a training data point

and take the average → $$\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i)$$

tally up all the times

where the model's classification was **different** from the true class

$n = 3$

Student	Grade	$C(X)$
Ab	B-	C+
Kaden	B	B
Kylee	A	D

$$Training\ error = \frac{1}{3}\sum_{i=1}^{3} I(y_i \neq \hat{y}_i)$$

$$= \tfrac{1}{3}\left((1) \quad + (0) \quad + (1) \quad \right)$$

$$= \frac{2}{3}$$

$$= 0.67$$

Takeaways

- Choosing the “right” level of flexibility is **critical** (in both regression and classification)
- Bias-variance trade off makes this challenging
- Coming up in Ch. 5:
  - Various methods for **estimating** test error rates
  - How to use these estimates to find the **optimal level** of flexibility

Code Distribution

GitHub

- We will use GitHub to distribute code, collect finished code, and facilitate pair programming

1. Create a GitHub account (https://github.com/)
2. Download GitHub Desktop

GitHub

- We will use GitHub to distribute code, collect finished code, and facilitate pair programming

1. Create a GitHub account (https://github.com/)
2. Download GitHub Desktop

Demo!

GitHub

- We will use GitHub to distribute code, collect finished code, and facilitate pair programming

1. Create a GitHub account (https://github.com/)
2. Download GitHub Desktop

- Practice accepting the in-class activity for today, modifying it, and updating your repository