The Bias-Variance Tradeoff

Objectives

- Describe bias and variance in machine learning
- Describe what it is to underfit and overfit data
- Relate underfitting and overfitting data to model bias and variance

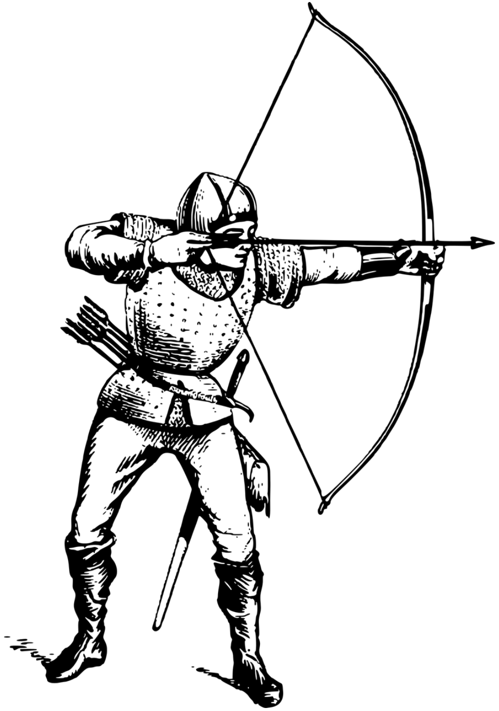
Common interview question:

What is the bias-variance trade-off?

You should able to answer this question at the end of this lecture, and related questions like:

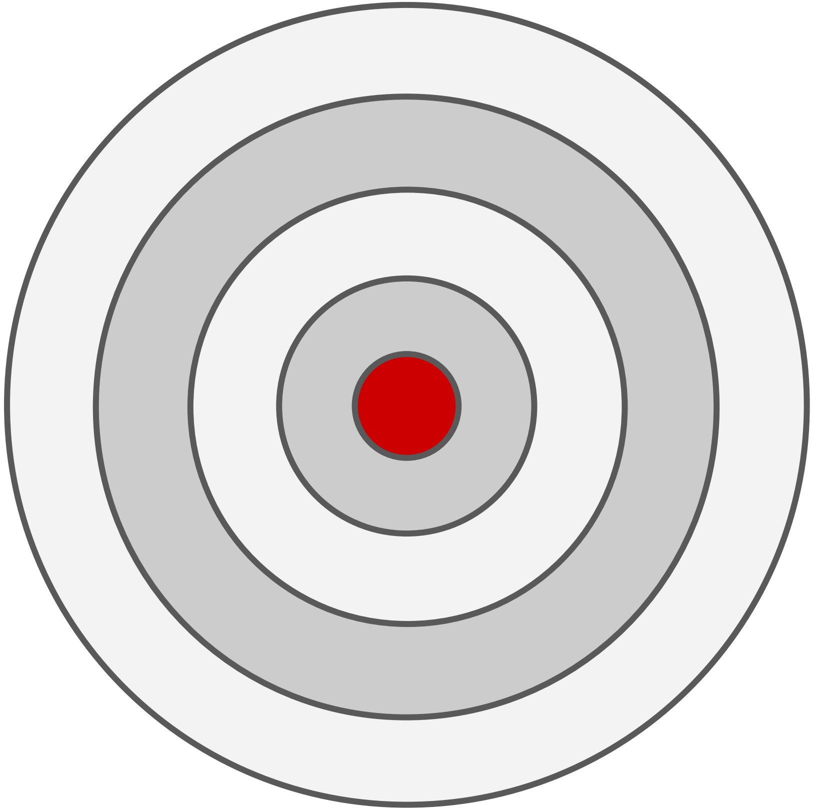
- Shouldn’t your model always perfectly predict your data?
- *Why* is there a bias-variance trade-off?

Bias and Variance Introduction



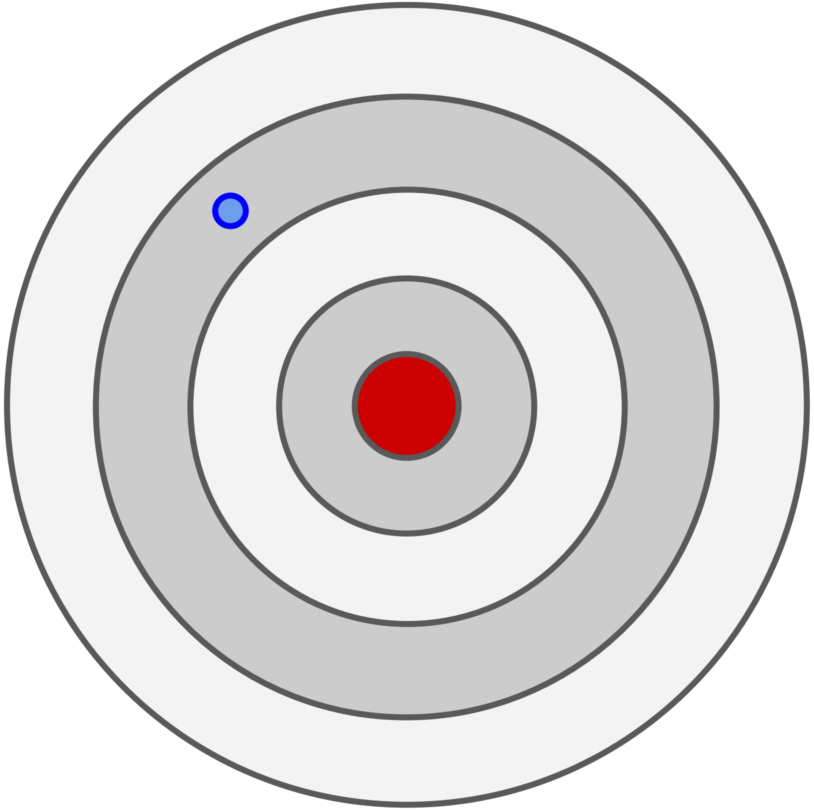
Bias and Variance Introduction

The archer takes aim and ...



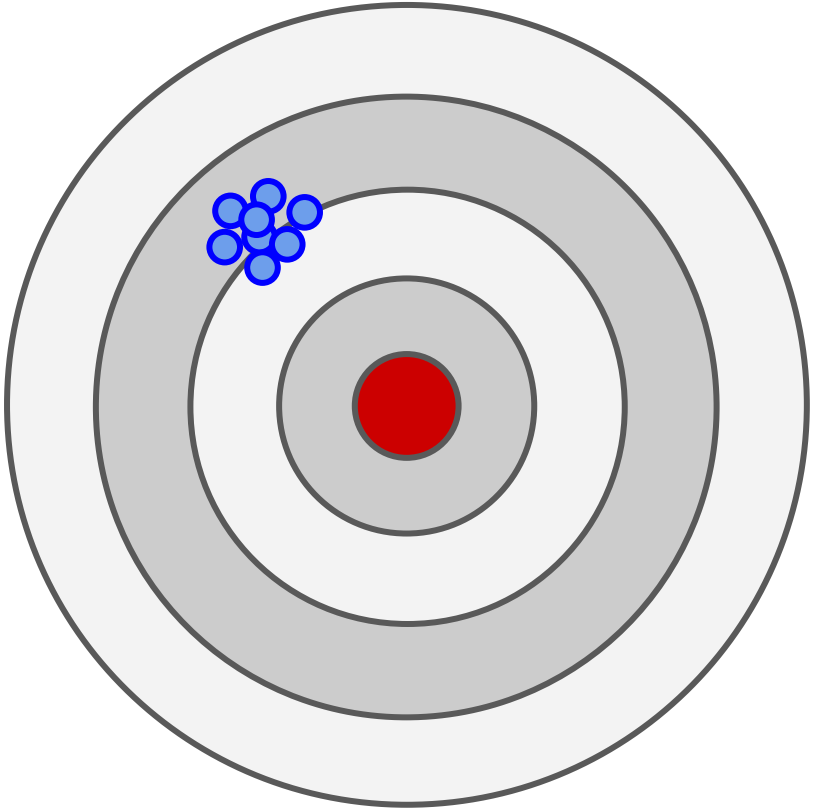
Bias and Variance Introduction

Releases...



Bias and Variance Introduction

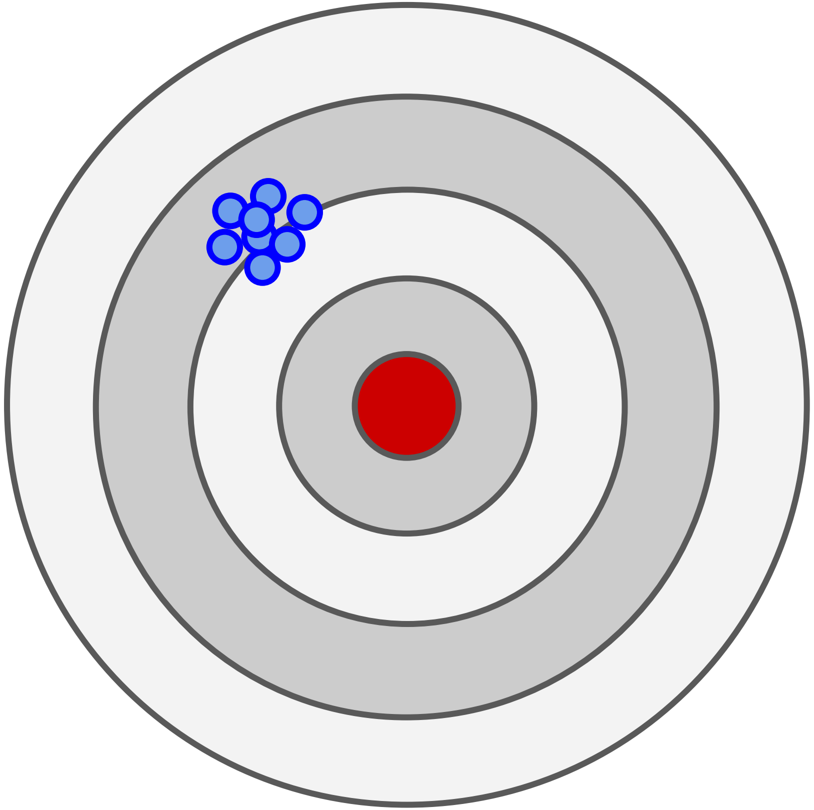
And then shoots 7 more ...



Bias and Variance Introduction

Statistical **bias**:

The amount the **expected value** of the results differ from the true value.



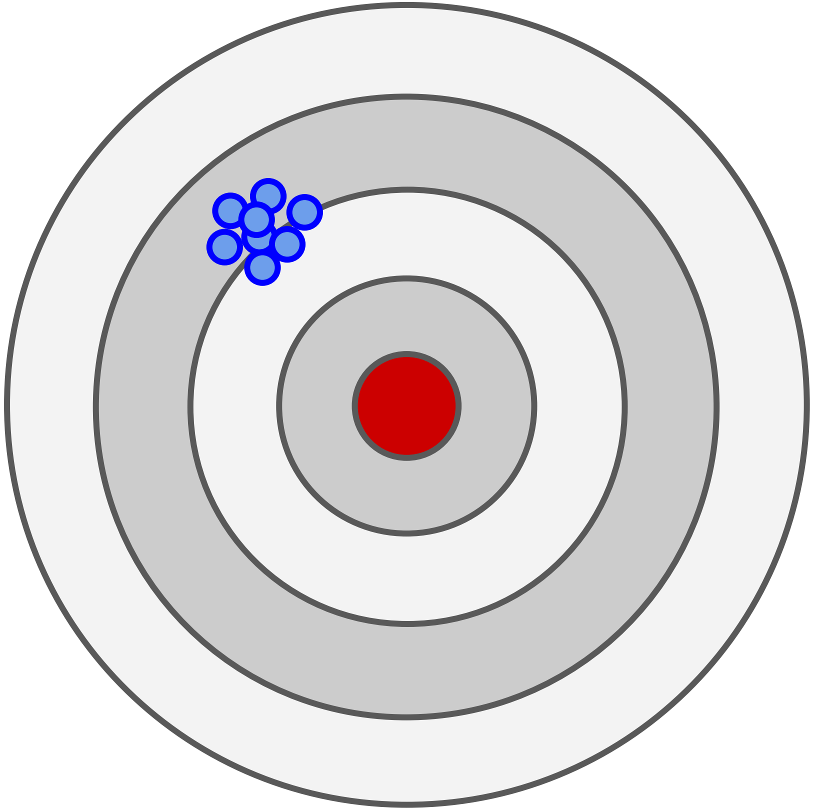
Bias and Variance Introduction

Statistical **bias**:

The amount the **expected value** of the results differ from the true value.

Expected value:

The **long-term average** value of the results of an experiment or function.



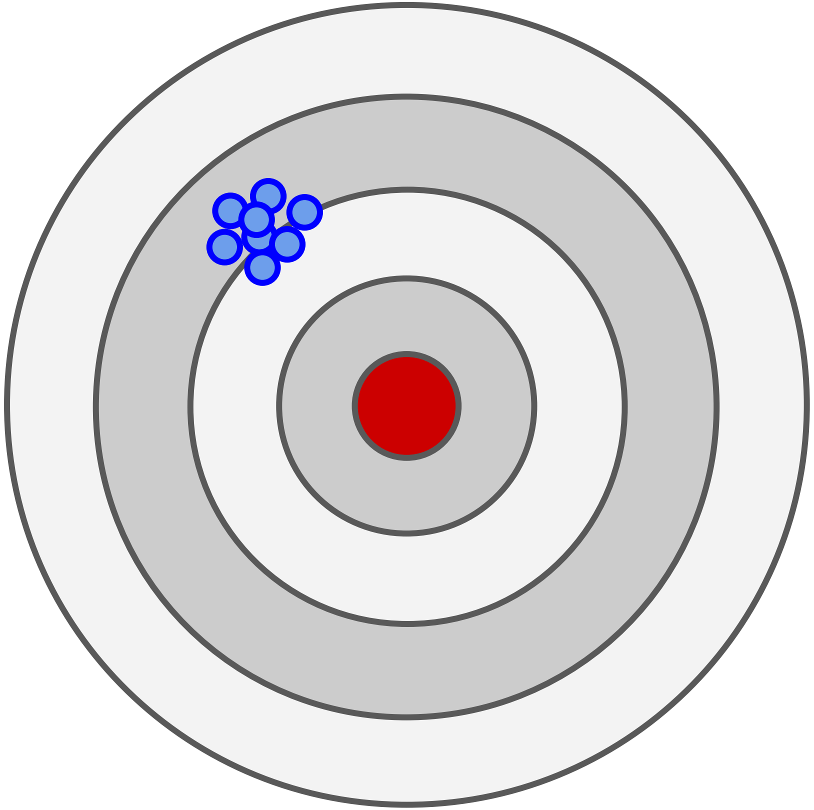
Bias and Variance Introduction

So in this archery analogy:

- What is the “true value”?

- What are the “results”?

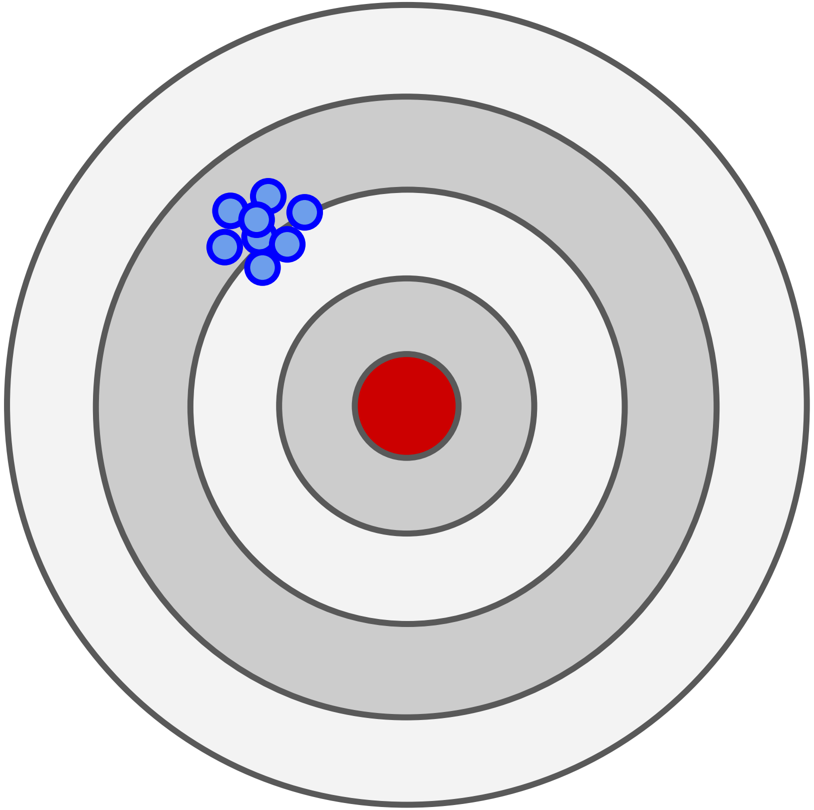
- What’s the model?



Bias and Variance Introduction

So in this archery analogy:

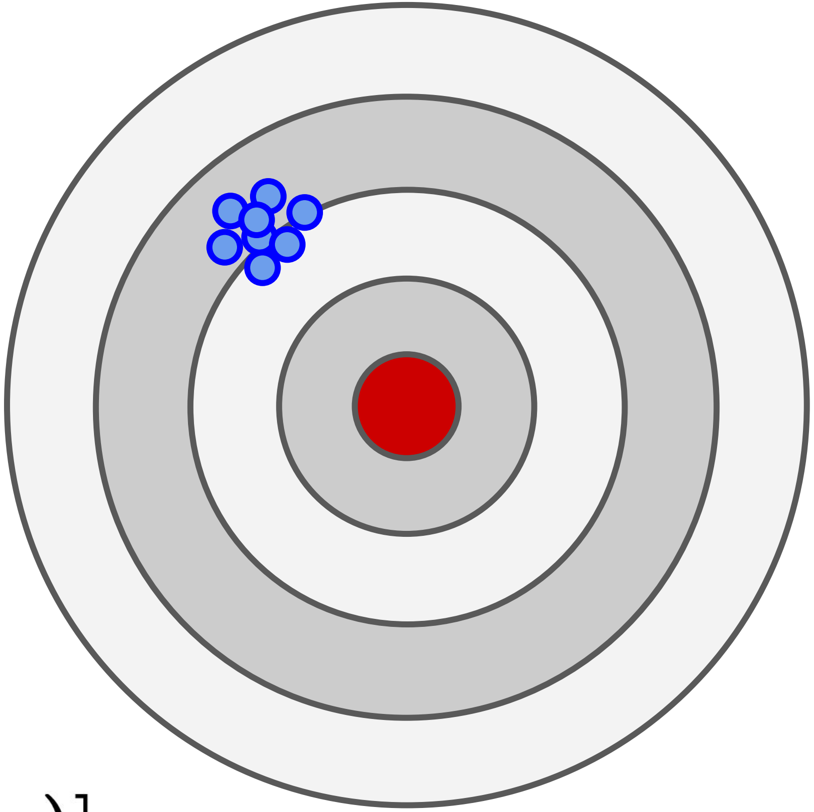
- What is the “true value”?
  the bullseye
- What are the “results”?
  where the arrows landed
- What’s the model?
  the archer



Bias and Variance Introduction

So in this archery analogy:

- What is the “true value”?
  the bullseye
- What are the “results”?
  where the arrows landed
- What’s the model?
  the archer



Adding some statistical notation:

true value: $f(x)$

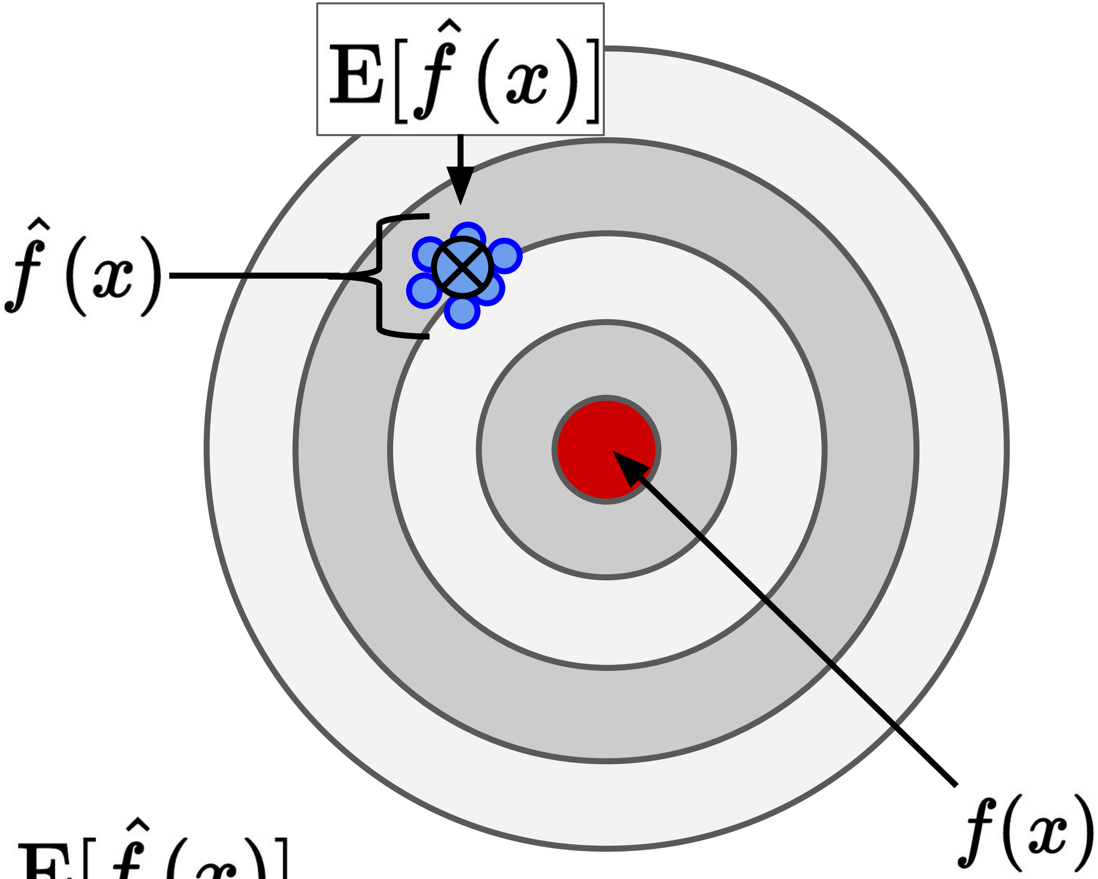
results: $\hat{f}(x)$

the expected value of the results: $\mathbf{E}[\hat{f}(x)]$

Bias and Variance Introduction

So in this archery analogy:

- What is the “true value”?
  the bullseye
- What are the “results”?
  where the arrows landed
- What’s the model?
  the archer



Adding some statistical notation:

true value: $f(x)$

results: $\hat{f}(x)$

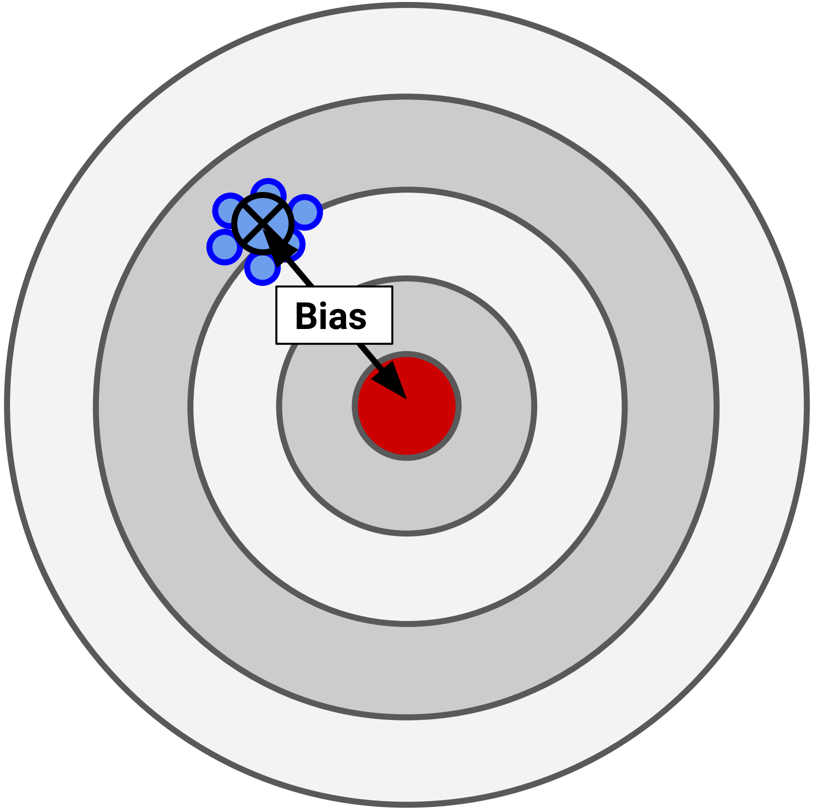
the expected value of the results: $\mathbf{E}[\hat{f}(x)]$

Bias and Variance Introduction

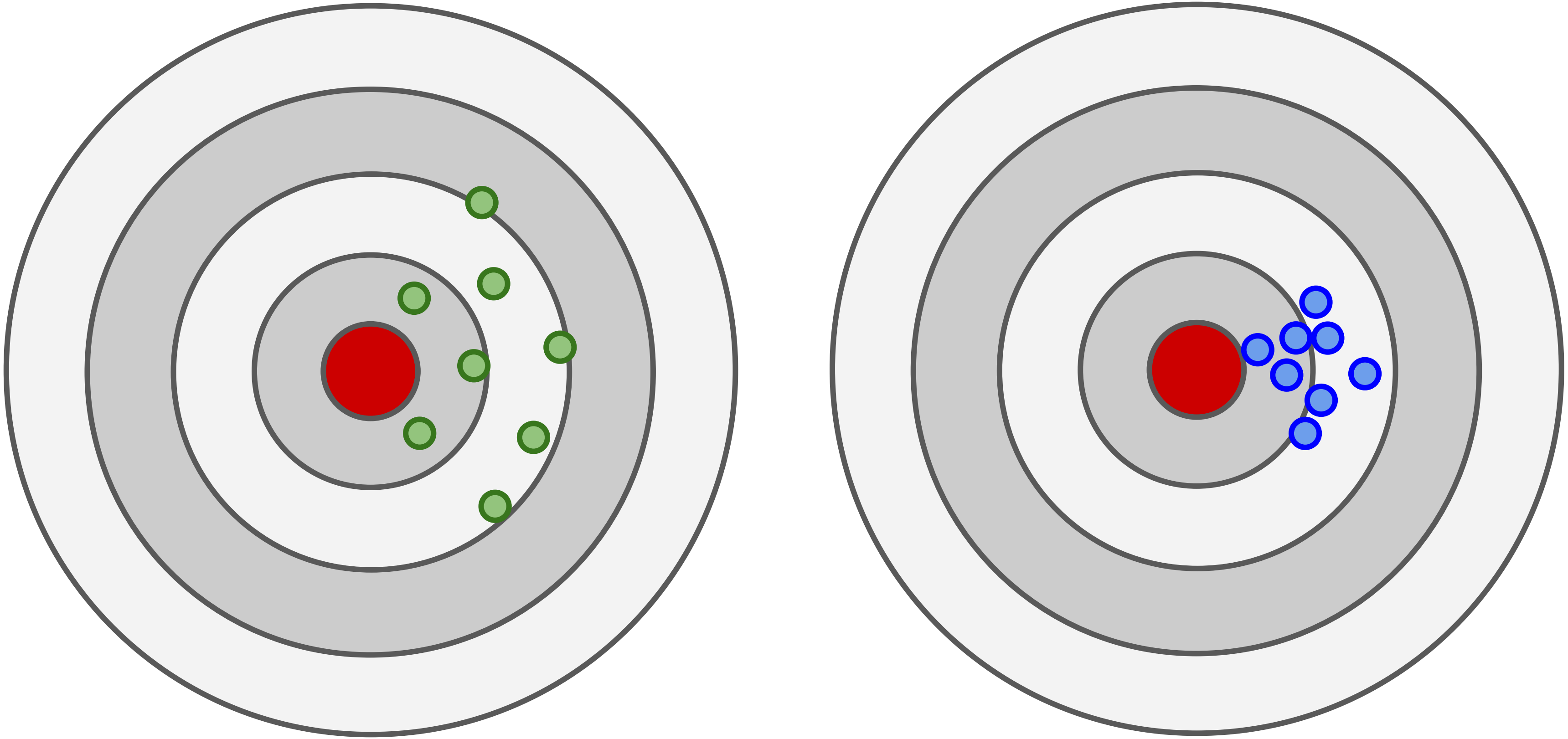
Bias:

The amount the **expected value** of the results differ from the true value.

$$\mathrm{Bias}\left[\hat{f}(x)\right] = \mathbf{E}\left[\hat{f}(x) - f(x)\right]$$

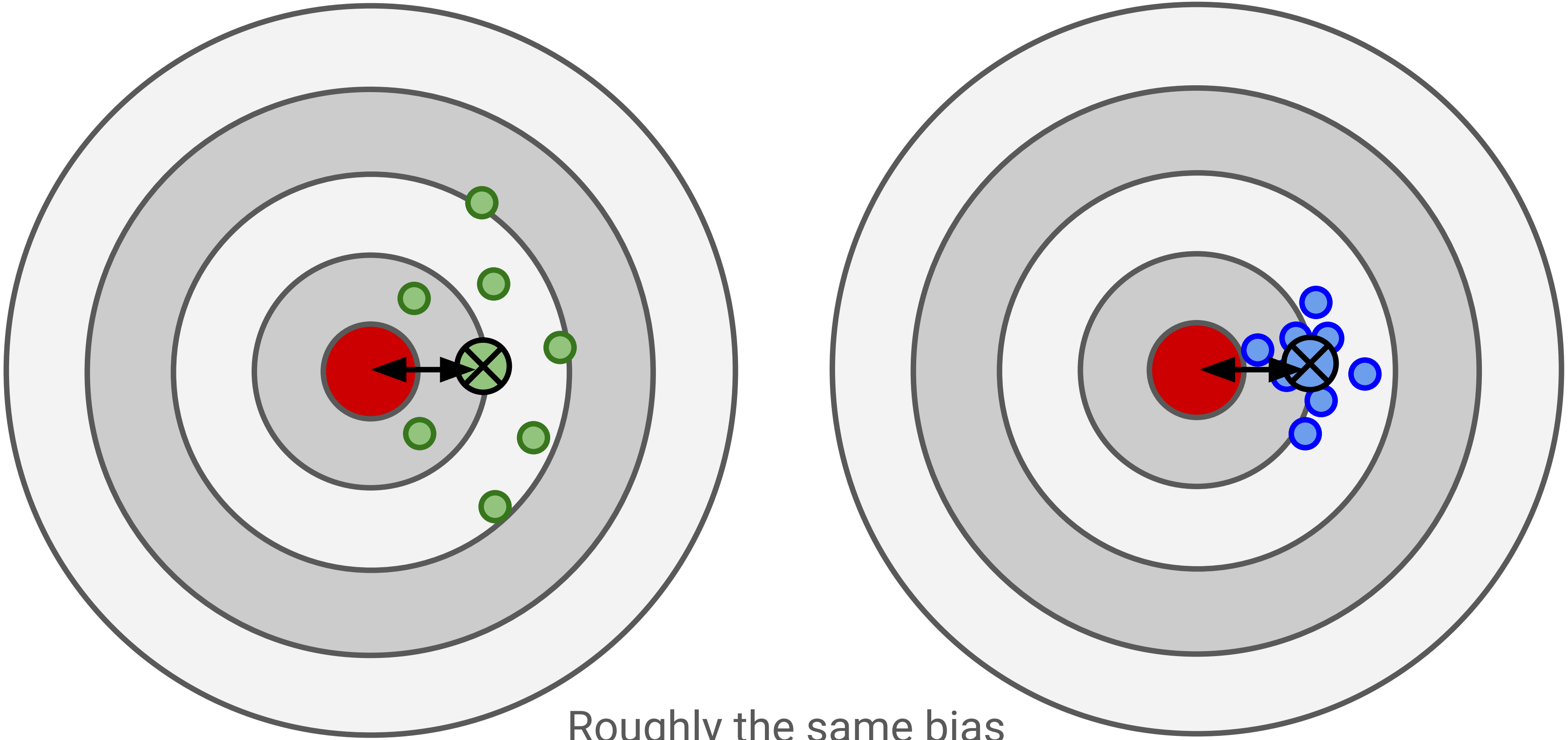


Bias and Variance Introduction



Compare the biases of these two archers (models)

Bias and Variance Introduction



Roughly the same bias

(the same value for the difference between the expected result and the true value)

Bias and Variance Introduction

Bias:

The amount the expected value of the results differ from the true value.

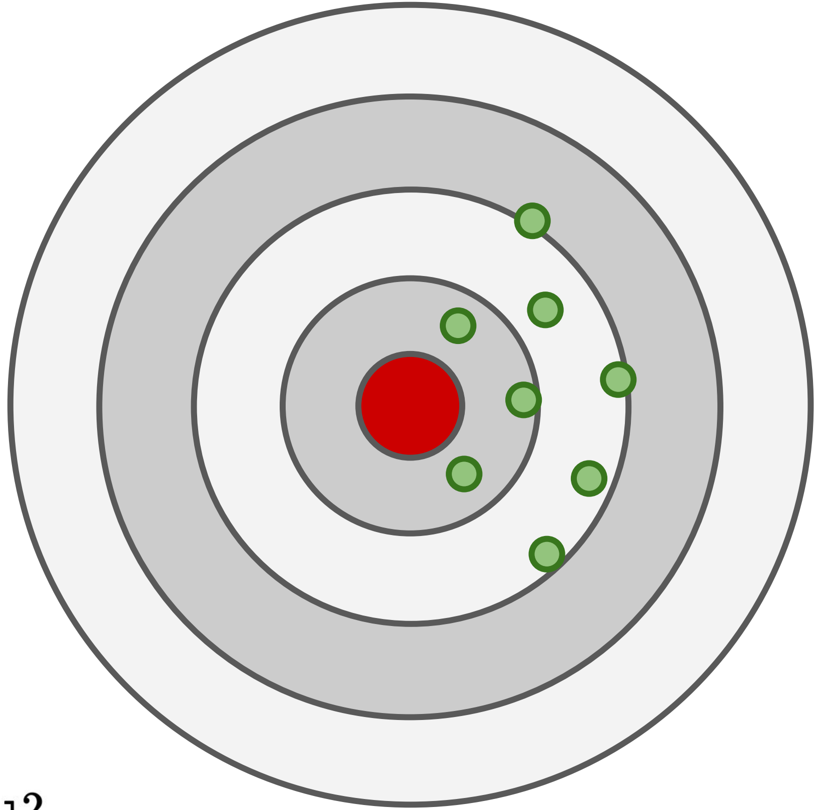
$$\mathrm{Bias}\left[\hat{f}\,(x)\right] = \mathbf{E}\left[\hat{f}\,(x) - f(x)\right]$$

Variance:

The expected value of the squared deviation of the results from the mean of the results.

$$\mathrm{Var}\left[\hat{f}\,(x)\right] = \mathbf{E}[\hat{f}\,(x)^2] - \mathbf{E}[\hat{f}\,(x)]^2$$

derivation

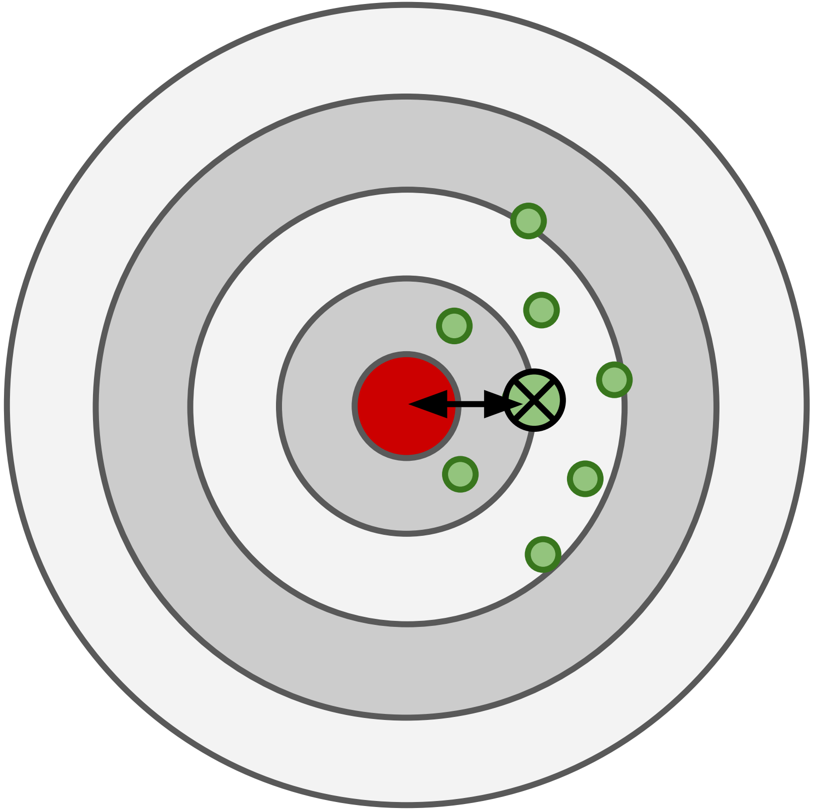


Bias and Variance Introduction

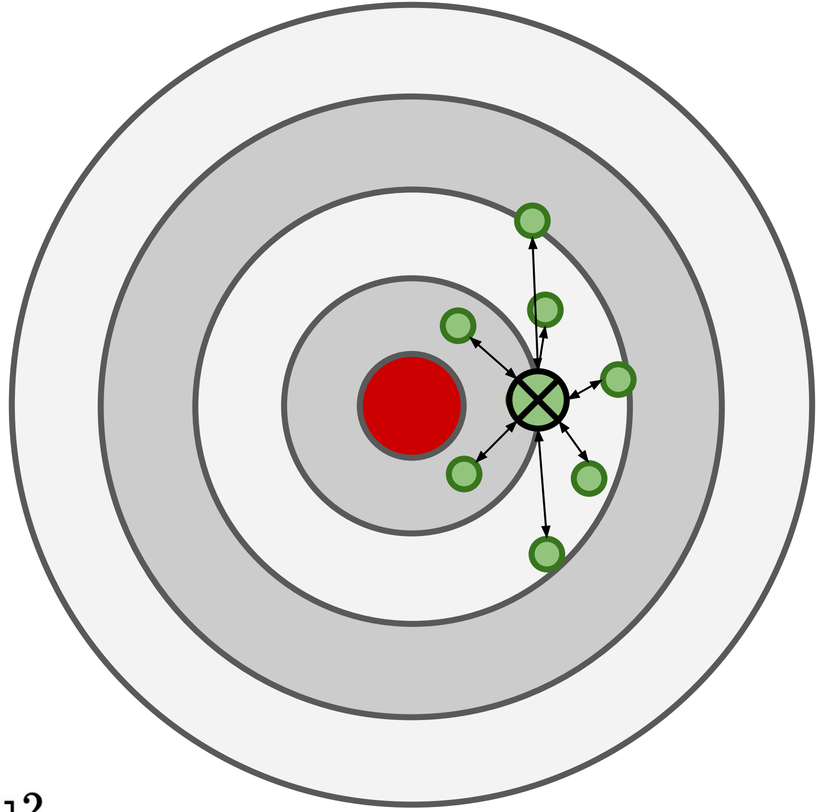
Bias:

The amount the expected value of the results differ from the true value.

$$\mathrm{Bias}\left[\hat{f}(x)\right] = \mathbf{E}\left[\hat{f}(x) - f(x)\right]$$



Bias and Variance Introduction



Variance:

The expected value of the squared deviation of the results from the mean of the results.*

$$\mathbf{Var}\left[\hat{f}(x)\right] = \mathbf{E}[\hat{f}(x)^2] - \mathbf{E}[\hat{f}(x)]^2$$

*no mention of true value!

Bias and Variance Intro - Exercise

In groups of two at your tables, draw archery results illustrating the following 4 scenarios:

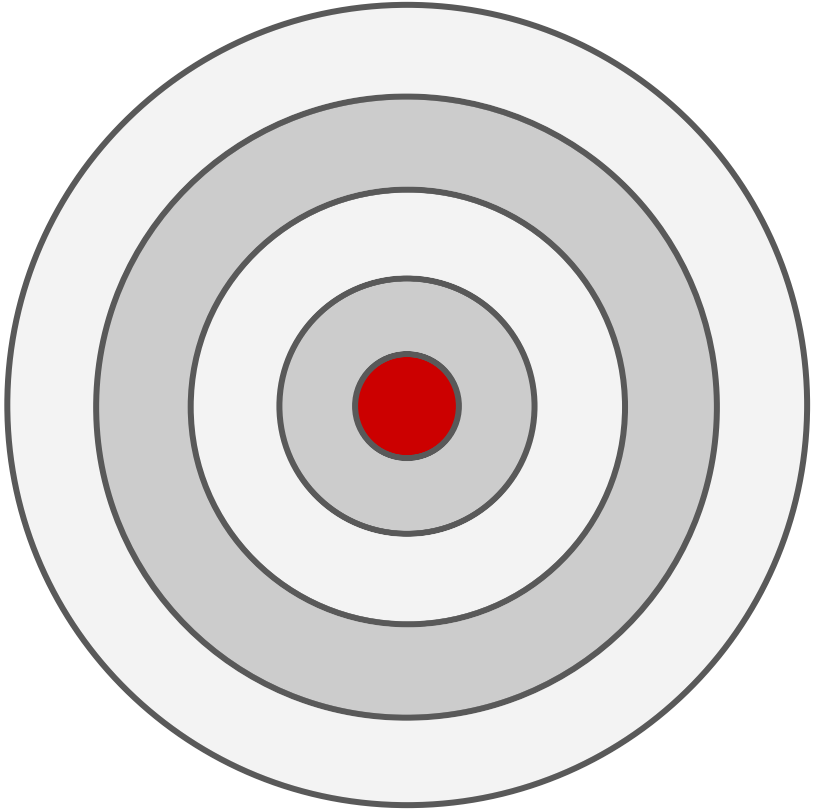
A) Low bias, low variance

B) Medium bias, high variance

C) High bias, low variance

D) Low bias, high variance

Who's the best archer?



Now some machine learning (will circle back to Bias & Variance in a bit)

(Supervised) Machine Learning

In supervised machine learning, **features** (a.k.a. predictors, exogenous variables, independent variables) are used to predict **targets** (a.k.a. endogenous variables, dependent variables).

Before the Kentucky Derby, let's say we have information from all races for the past year. What are the features? What are the targets?

And why are we doing this?

(Supervised) Machine Learning

In supervised machine learning, we’d like to train a model on past data, and then predict the outcome on new data.

For instance, given the horses, the race conditions, the jockeys - we’d like to predict the winner of the Kentucky Derby before it happens.

And so we’d like to have a very good model. How to do that?

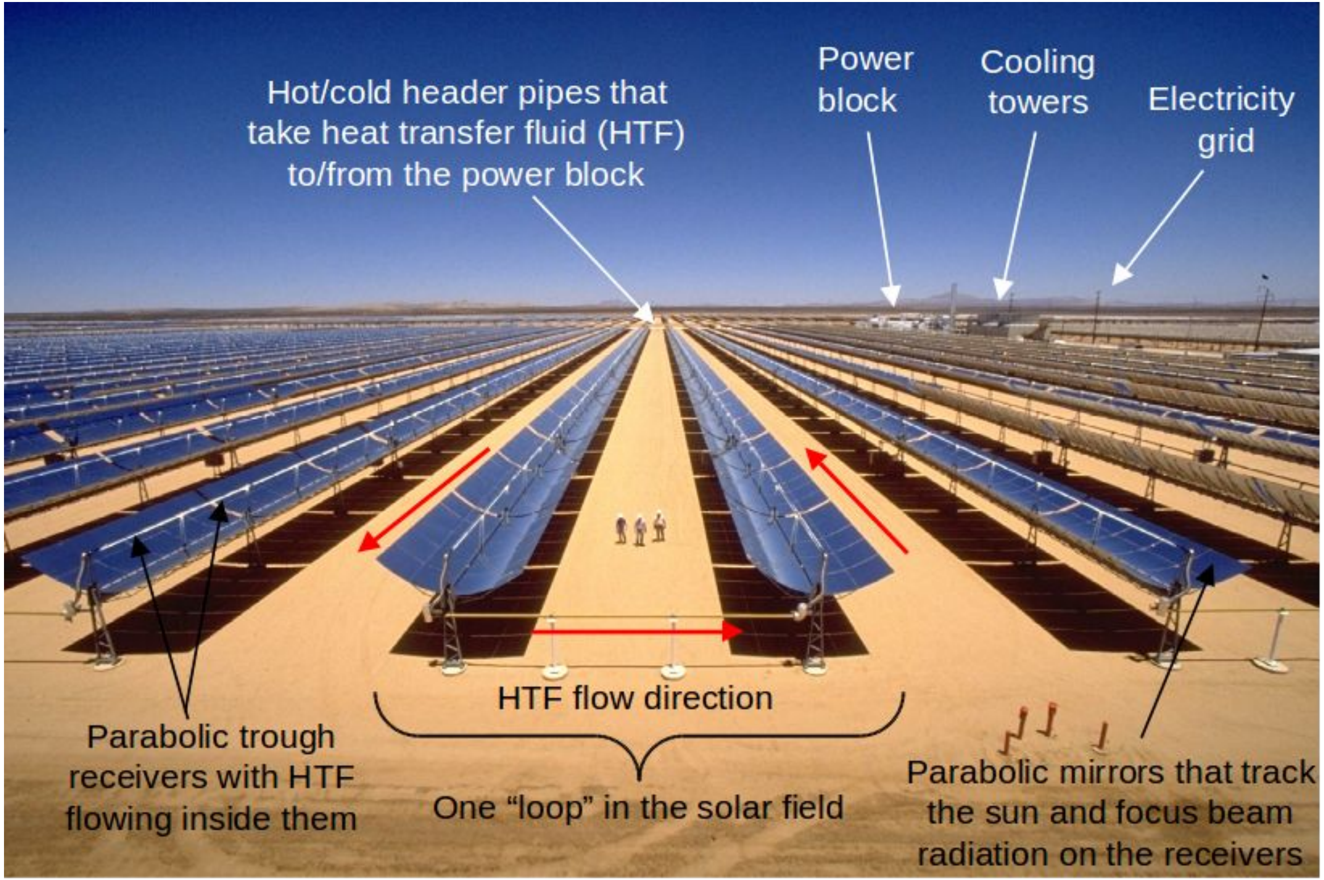
“Solving” Data Science - let’s perfectly fit the data

```
def train_super_awesome_perfect_model (X, y):
    while True:
        model = LinearRegression()
        model.fit(X, y)
        if calculate_r2(model, X, y) >= 0.999:
            return model
        else:
            X = insert_random_interaction_feature(X)
```

This approach will eventually perfectly fit that data we have.
Valiant but naive. Let me explain why this is a bad idea....

credit Ryan Henning for amazing function

Signal and noise (from my Ph.D.)



Hot/cold header pipes that take heat transfer fluid (HTF) to/from the power block

Power block

Cooling towers

Electricity grid

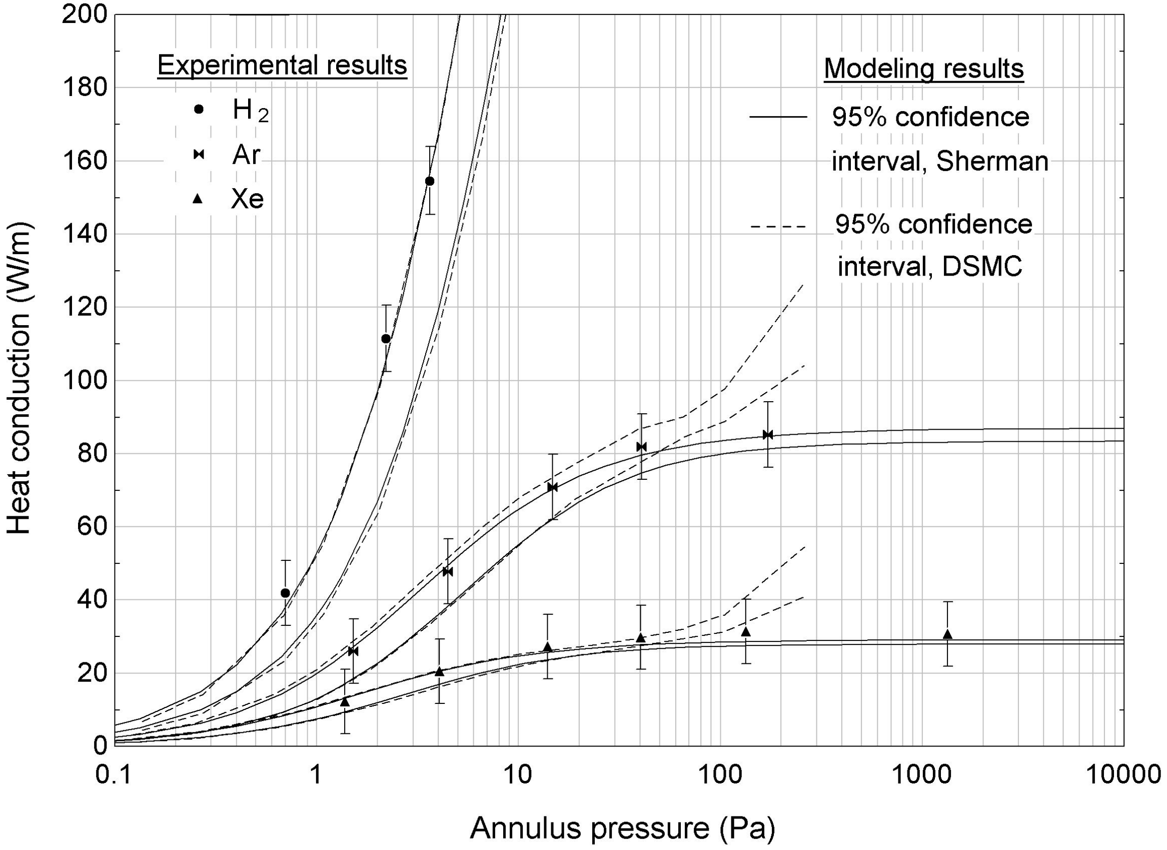
HTF flow direction

Parabolic trough receivers with HTF flowing inside them

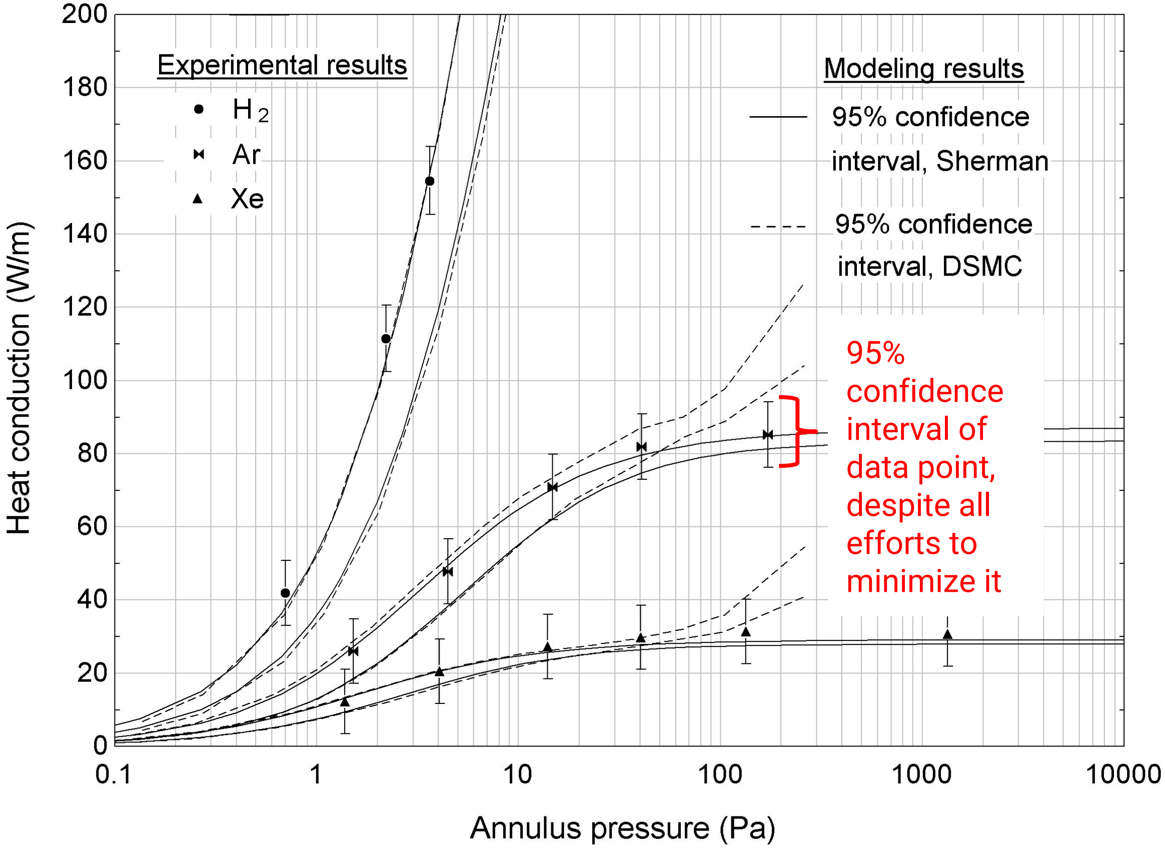
One “loop” in the solar field

Parabolic mirrors that track the sun and focus beam radiation on the receivers

Results (in a controlled laboratory environment)



Results (in a controlled laboratory environment)



95% confidence interval of data point, despite all efforts to minimize it

Most data is a combination of:

- signal (that we want to model)
- random noise (that we don't)

Not knowing either of these quantities exactly affects our ability to make good predictive models.

Unknown signal & noise leads to possible:

Underfitting: The model doesn't fully capture the relationship between predictors and the target in the data.

The model has *not* learned the data's underlying signal or true value.

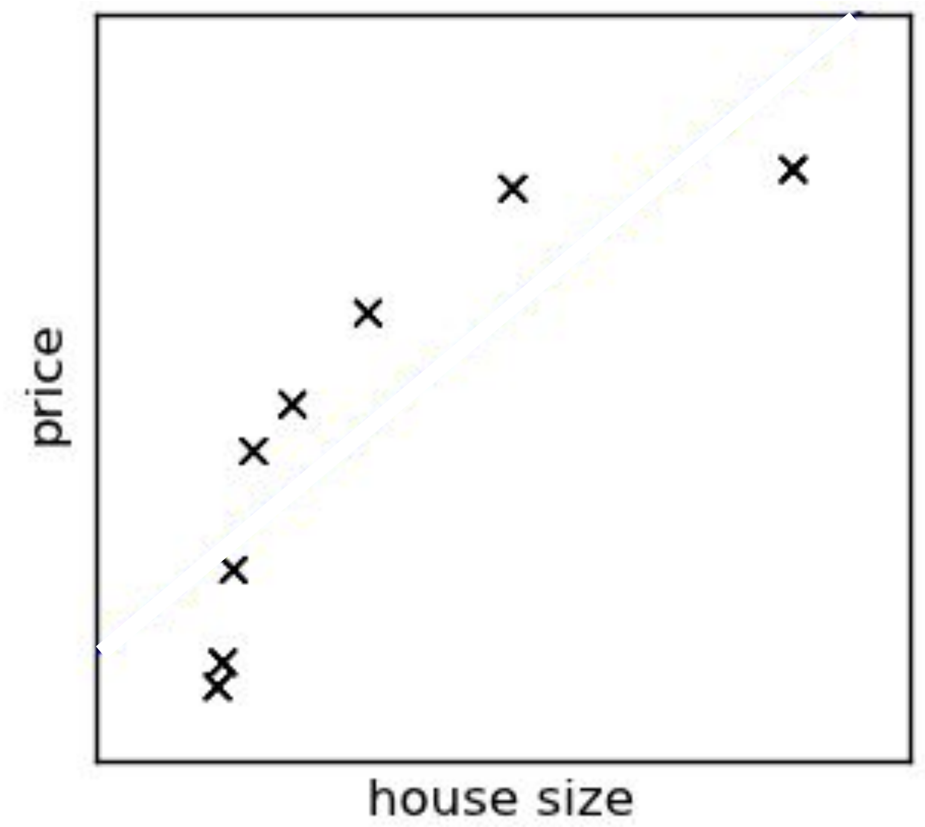
Overfitting: The model has captured noise in the data.

The model has learned the data's signal *and* the noise.

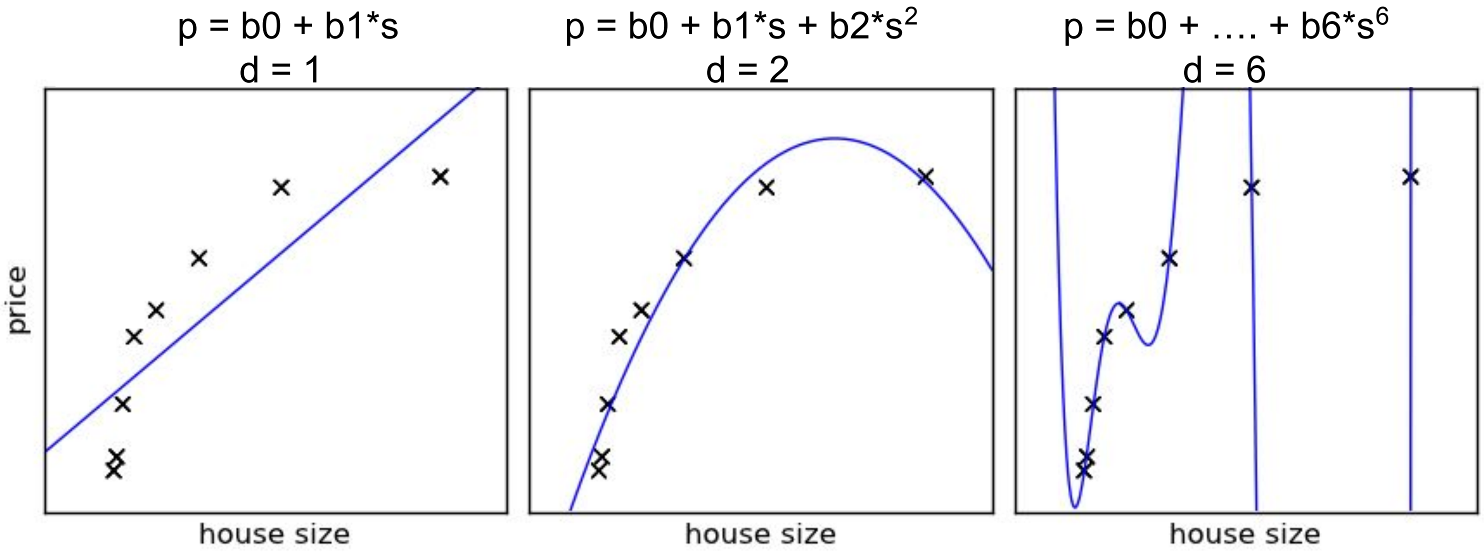
Example - let's fit this data

What do you think the signal is?

In other words, how do you think house price is related to house size?



Example - let’s fit this data



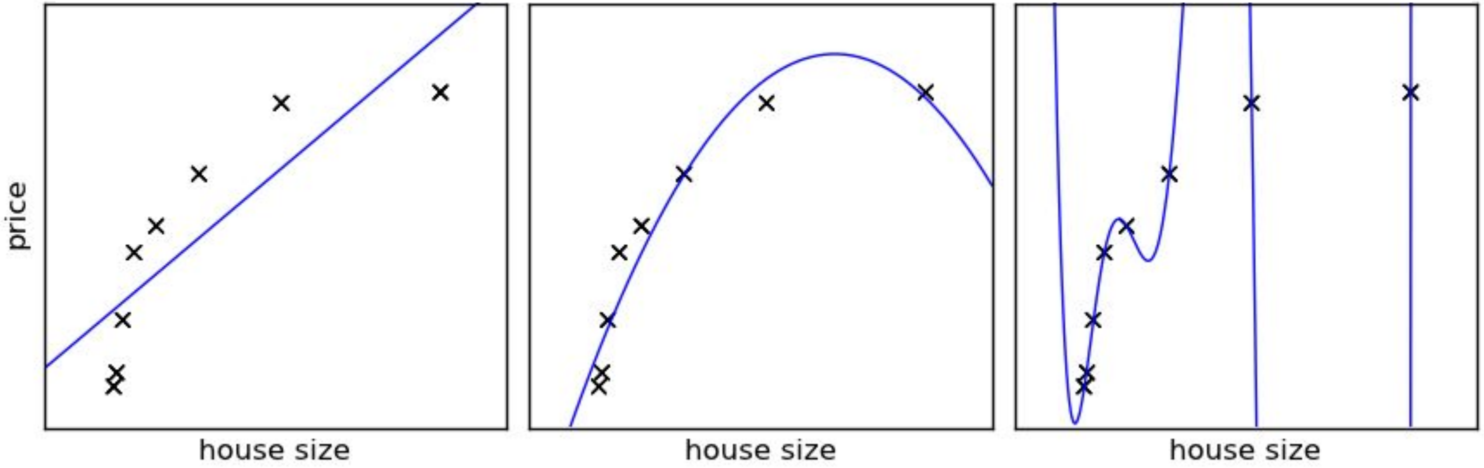
$p = b0 + b1*s$
$d = 1$

$p = b0 + b1*s + b2*s^2$
$d = 2$

$p = b0 + …. + b6*s^6$
$d = 6$

We’ve decided to use a polynomial as the functional form of the signal...

Example - let’s fit this data



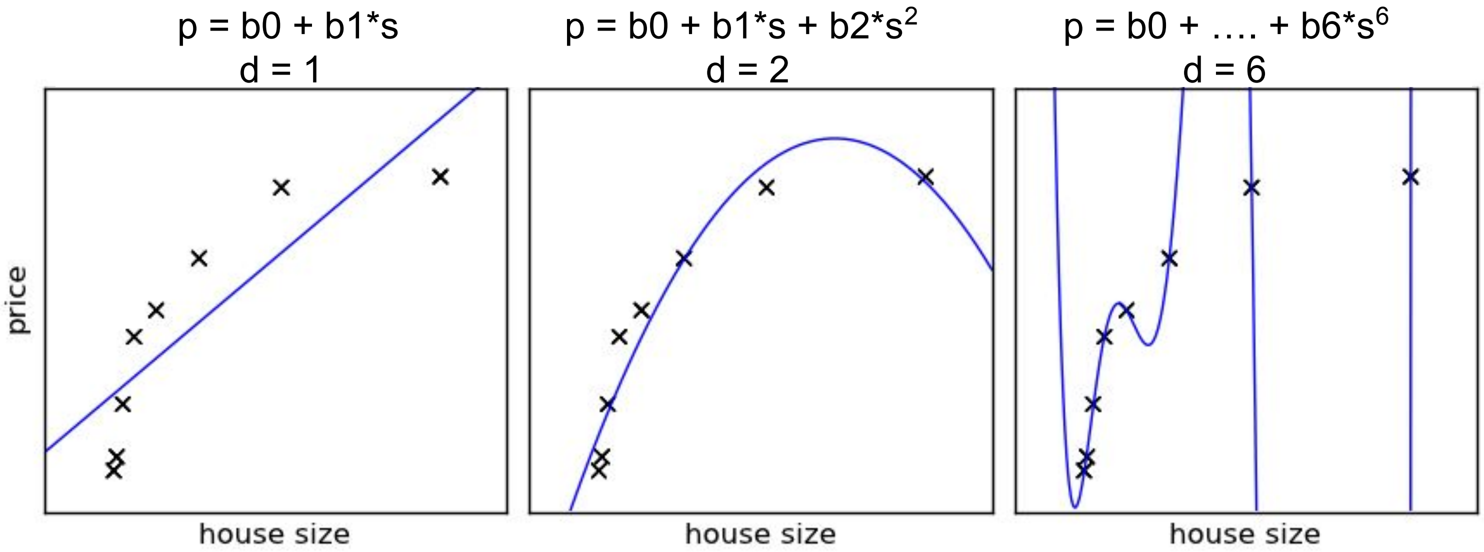
$p = b0 + b1*s$
$d = 1$

$p = b0 + b1*s + b2*s^2$
$d = 2$

$p = b0 + ... + b6*s^6$
$d = 6$

First order fit. Underfit, properly fit, overfit? Why?

Example - let’s fit this data



$p = b0 + b1*s$
d = 1

$p = b0 + b1*s + b2*s^2$
d = 2

$p = b0 + …. + b6*s^6$
d = 6

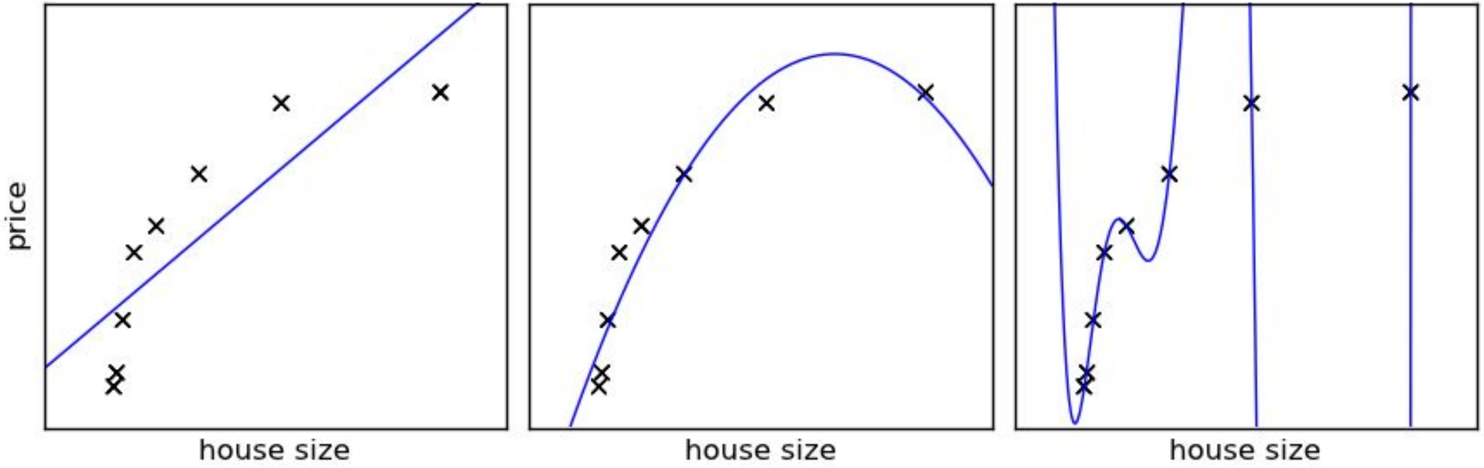
Likely underfit.
The fit doesn’t fully capture the relationship between house size and price in the data.

Example - let’s fit this data

$p = b0 + b1*s$
$d = 1$

$p = b0 + b1*s + b2*s^2$
$d = 2$

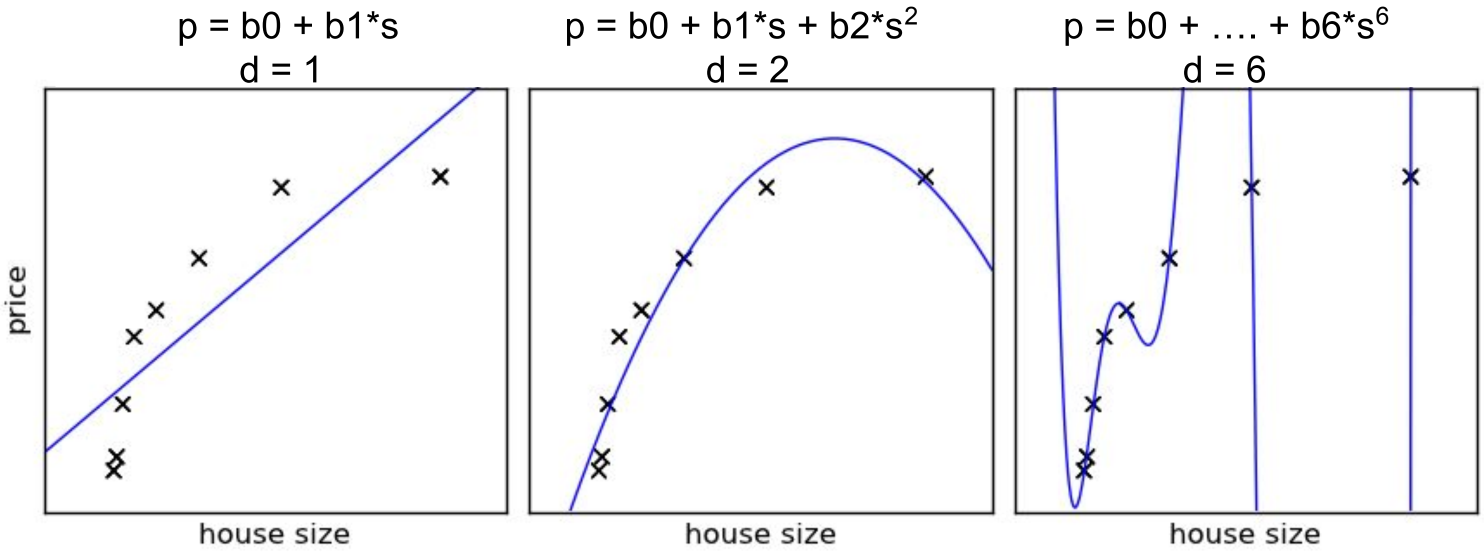
$p = b0 + ... + b6*s^6$
$d = 6$



Likely underfit. The fit doesn’t fully capture the relationship between house size and price in the data.

Sixth order fit. Underfit, properly fit, overfit? Why?

Example - let’s fit this data



p = b0 + b1*s
d = 1

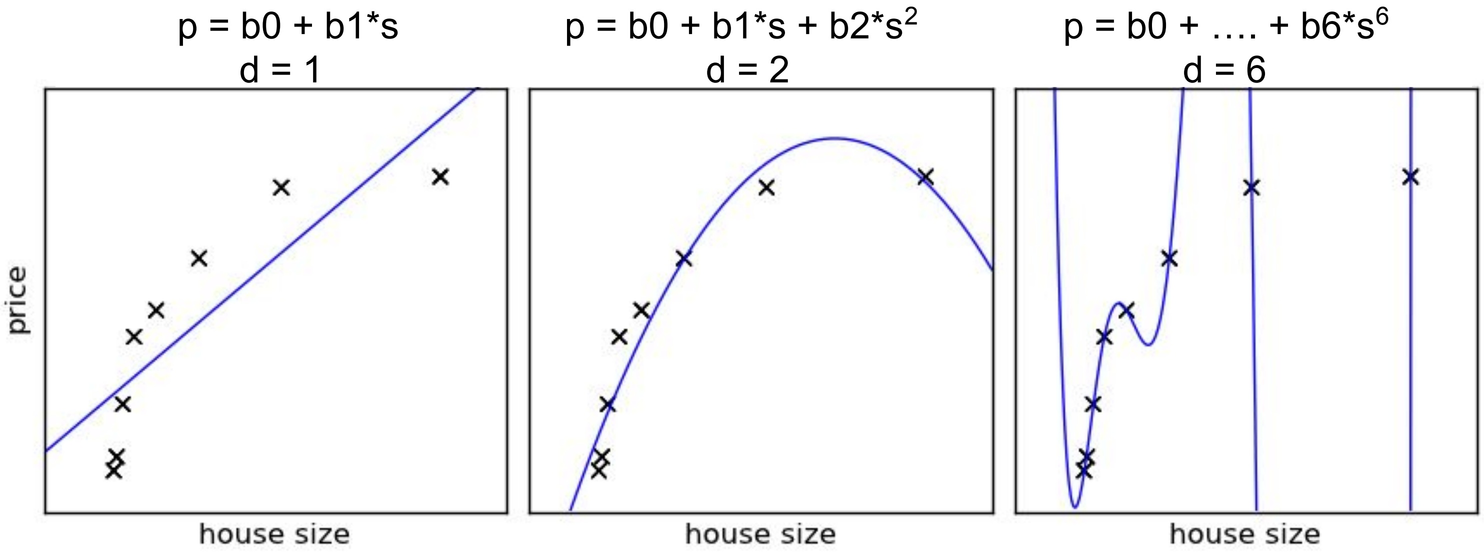
p = b0 + b1*s + b2*s^2
d = 2

p = b0 + …. + b6*s^6
d = 6

Likely underfit.
The fit doesn’t fully capture the relationship between house size and price in the data.

Likely overfit.
The fit perfectly models the data, including noise in the data.

Example - let’s fit this data



$p = b0 + b1*s$

$d = 1$

$p = b0 + b1*s + b2*s^2$

$d = 2$

$p = b0 + …. + b6*s^6$

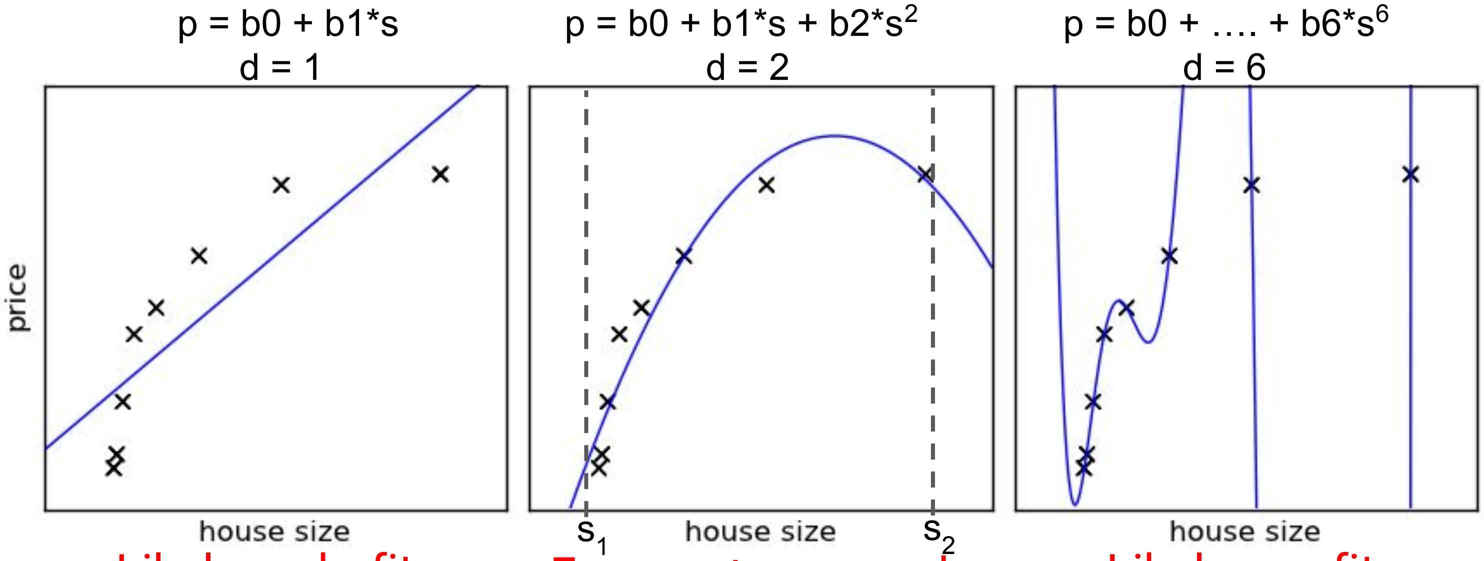
$d = 6$

Likely underfit. The fit doesn’t fully capture the relationship between house size and price in the data.

Second order fit. Underfit, properly fit, overfit? Why?

Likely overfit. The fit perfectly fits the data, including noise in the data.

Example - let’s fit this data



$p = b0 + b1*s$

$d = 1$

$p = b0 + b1*s + b2*s^2$

$d = 2$

$p = b0 + …. + b6*s^6$

$d = 6$

Likely underfit. The fit doesn’t fully capture the relationship between house size and price in the data.

From s_1 to s_2, maybe properly fit*

*but what about $>s_2$?

Functional form flawed (see logarithmic)

Likely overfit. The fit perfectly fits the data, including noise in the data.

Evidence on the impact of sustained exposure to air pollution on life expectancy from China's Huai River policy

12936–12941 | PNAS | August 6, 2013 | vol. 110 | no. 32

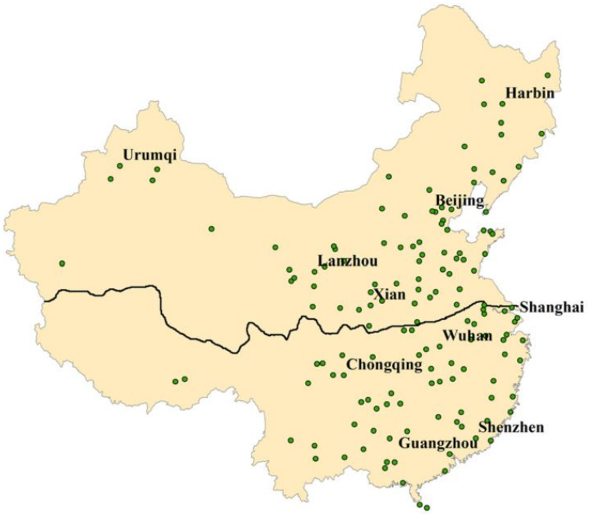
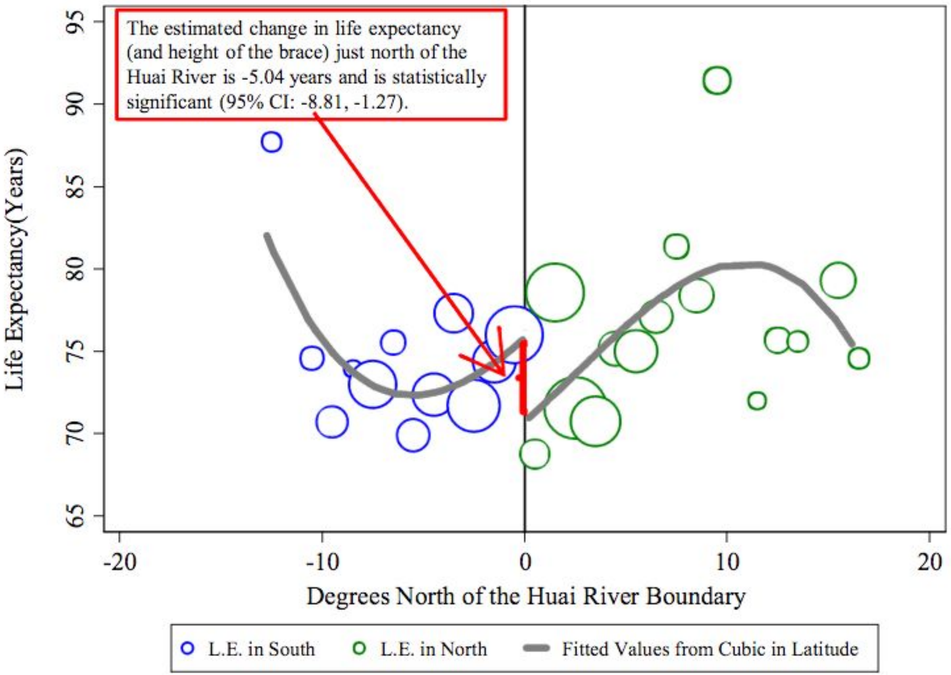


Fig. 1. The cities shown are the locations of the Disease Surveillance Points. Cities north of the solid line were covered by the home heating policy.



The estimated change in life expectancy (and height of the brace) just north of the Huai River is -5.04 years and is statistically significant (95% CI: -8.81, -1.27).

In groups of four at your table, discuss: underfit, properly fit, overfit? Why?

Back to
Bias and Variance in Machine Learning

Model performance in terms of Bias and Variance

The goal of supervised predictive models is to predict accurately on unseen data.

We assume the true predictor/target relationship is given by an unknown function plus some sampling error, ε, that has zero mean and σ^2 variance

$$y = f(x) + \epsilon$$

We estimate the true function by fitting a model over the observed data.

$$\hat{y} = \hat{f}(x)$$

The error of the model on unseen data x, y can be decomposed into:

$$\mathrm{E}\left[\left(y - \hat{f}(x)\right)^2\right] = \mathrm{Bias}\left[\hat{f}(x)\right]^2 + \mathrm{Var}\left[\hat{f}(x)\right] + \sigma^2$$

derivation and explanation on Wikipedia

The Bias-Variance Trade-off

See jupyter notebook:

demo_bias_variance_tradeoff.ipynb

Verbalizing the Bias-Variance Trade-off

A low bias model accurately predicts the population’s underlying true value, or signal, and vice - versa.

A low variance model’s predictions don’t change much when it is fit on different data from the underlying population (and vice-versa).

A trade-off often exists between bias and variance because some amount of model complexity is often required to match the underlying population signal, but this same complexity also makes the fit more sensitive to variations in the data the model is fit on. So as bias decreases, variance often increases (and vice-versa).

Discuss with a partner

An aspiring data scientist comes up to you and says “A high variance model is simply overfitting, and a high bias model is just underfit.” What do you think?

(By the way - this simple thinking is how you *don’t* get hired as an instructor here.)

With B/V trade-off: Picking the Best Model

Quick break

Then we’ll get into Cross-Validation and how it’s used to find the complexity sweet spot.